1973

Description and FORTRAN simulation of MK 86 AA ballistics.

Kidd, Delbert Ray.

Monterey, California. Naval Postgraduate School

DESCRIPTION AND FORTRAN SIMULATION OF MK 86 AA BALLISTICS

Delbert Ray Kidd

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESCRIPTION AND FORTRAN SIMULATION
OF
MK 86 AA BALLISTICS

by

Delbert Ray Kidd

Thesis Advisor: J. R. Ward

June 1973

Approved for public release; distribution unlimited.

Description and FORTRAN
Simulation of MK 86 AA
Ballistics

by

Delbert Ray Kidd
Lieutenant, United States Navy
B.S., University of New Mexico, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1973

ABSTRACT

The GFCS MK86 solves the ballistics problem for the 5"/54 gun iteratively utilizing table look-ups. Each ballistics parameter table is sub-divided two-dimensionally, corresponding to selected elevation and slant range intervals. The final value for each parameter is found by interpolating between parameter values evaluated at elevation angles which bracket the actual elevation angle.

This thesis describes the method of determining the ballistics solution for AA cartesian ballistics. A description of the ballistics geometry and of the model utilized is presented, together with a FORTRAN simulation of the MK86 computer program and a listing of the ballistics tables used in this system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL DESCRIPTION

The GFCS MK86 is the first digital gun-fire control system to be considered by the U. S. Navy. It is designed to control the 5"/54 gun-mount and to solve the fire control problem for the air, surface, shore-direct or shore-indirect modes of operation. In this thesis only the AA cartesian ballistics mode is considered.

B. PROBLEM DESCRIPTION

A computer simulation program was felt to be a prerequisite to a complete study and analysis of the GFCS MK86 AA model This in turn necessitated a careful study of the existing documentation and machine-language software of the MK86 system.

The FORTRAN simulation of the MK 86 AA mode was divided into three sections: The pre-filter processing section, programmed by LT J. D. GORMAN; The three-dimensional numerical filter, by LT T. J. WILL; and the ballistics section contained in this thesis. These three programs can process radar target information from target detection to ballistics output prior to the generation of gun orders.

Another objective of this project was the generation of adequate documentation of the system. This phase was also divided into the three sections described in the previous paragraph.

C. SUMMARY OF INTRODUCTION

This thesis contains a description of the MK86 ballistics program including the ballistics intercept geometry, ballistics model used, and sequence of ballistics computations. A description of each subprogram indicating its function is also presented. The FORTRAN simulation program including the ballistics tables is listed and differences between the machine-language program and the FORTRAN program are indicated.

II. INTRODUCTION TO MK86 BALLISTICS

The GFCS MK86 ballistics program accepts stabilized and filtered target position and rate data, together with ownship's data, and computes the gun ballistics required for the generation of gun pointing and fuze orders.

A. BALLISTICS SETUP (I. E. INITIALIZATION)

The ballistics parameters such as air temperature, air density, wind, and initial velocity are entered at the console and corresponding increments from noninal values are computed.

Target and ownship's data are used to compute the target's present position and rate relative to the gun's position at ballistics time. This position of the target is called point 1 (see Figure 1), and is computed in spherical coordinates. The relative rates are resolved into range and crossrange components.

A test is made to determine whether the input data is associated with a new target. If it is, an acquisition mode is declared and the processing sequence calls the acquisition subprogram. In the case of an old target, track mode is designated and the acquisition subprogram is bypassed. This is illustrated in the flow chart of Figure 2.

B. BALLISTICS ACQUISITION

Before a ballistics solution can be developed, a first estimate of future target position is required. This is called the advance position, or point 3 (see Figure 1). If the system has been tracking the target, then the last ballistics solution, updated, serves as the advance position. With a new target, however, some other means for estimating a reasonably accurate point 3 is necessary. This is the function of the ballistics acquisition subprogram (see Figure 2).

In the acquisition phase, point 3 is computed as follows. The projectile time of flight to the present target position (point 1) is computed from the ballistics table in the normal way (see Section III). Using this time of flight and the target's present position and velocity, an estimate is made of the target's future position (point 3). A number of logical tests are employed to check the validity of the estimate. The slant range, bearing, and elevation of point 3 are designated R3, BY3, and E3 respectively.

C. DETERMINATION OF POINT 2

The ballistics program next uses the advance position of the target (point 3) as the aim point and computes a predicted "impact" point from the ballistics tables. This is called point 2 and the corresponding slant range, bearing, elevation, and time of flight are designated R2, BY2, E2, and T2, respectively. These values differ

from the point 3 quantities due to non-nominal powder and air temperatures, and because of coriolis, drift, and wind effects.

D. CALCULATION OF POINTS 21 AND 22.

The target position which corresponds to the computed time of flight T2 is calculated and designated as point 21 (see Figure 1). Then the differences in slant range, bearing and elevation from those at point 2 (the predicted projectile impact point) are computed. These increments are designated IR21, IBY21, and IE21, respectively, and a corresponding linearized estimate of the time of flight increment IT2 is computed.

Finally the point 21 target position is updated by the time increment IT2 to determine point 22 (see Figure 1), which is the first estimate of the point where the target and projectile will coincide in both position and time.

E. BALLISTICS ITERATION (I. E. SECOND PASS)

After point 22 has been calculated once, as indicated in the preceding sections, the program uses point 22 (modified) as a new (improved) point 3 and goes through the calculation to point 22 a second time. This is shown in Figure 2. Finally the increments IR22, IBY22, IE22, and IT22 between point 2 and point 22 are computed for use in the ballistics output program.

F. CLOSURE TESTS

A test is made at the end of each of the two passes described above to determine if the final point 22 is a valid (i. e. accurate) solution. This is done by comparing point 21 data and point 2 data, which is called the closure test. The test after the first pass merely sets a flag, which is used to modify the criterion used in the test which follows the second pass. If the second-pass solution fails to close, then the computation aborts for this time interval.

G. BALLISTICS OUTPUT

If closure has occurred, the gun elevation, bearing and time of flight which were calculated with point 2 as the aim point are modified by the amount necessary to move the impact point to point 22. This data is then transmitted to the gun and fuze order programs.

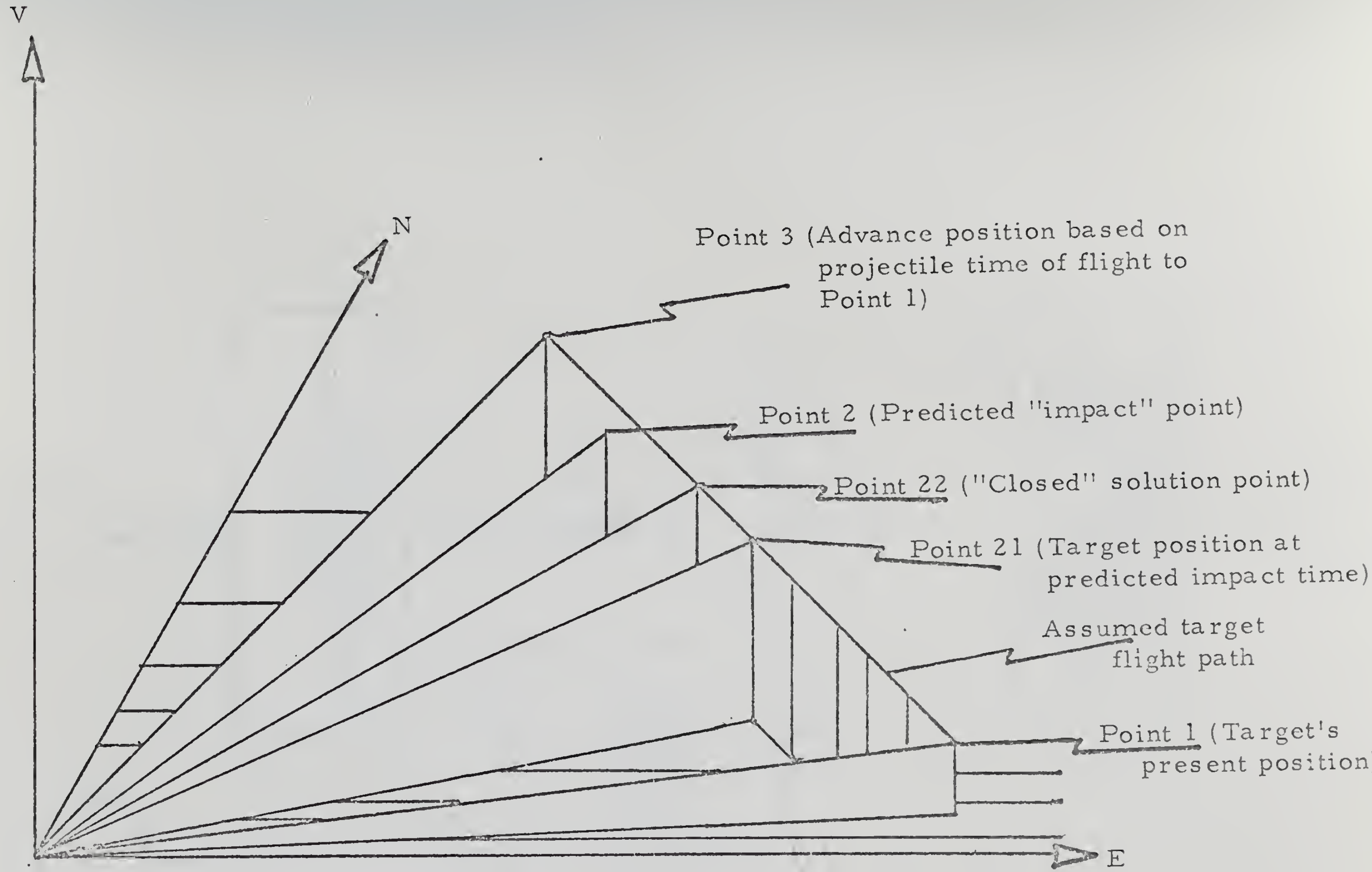

Figure 1

GFCS MK 86 BALLISTICS CLOSURE GEOMETRY

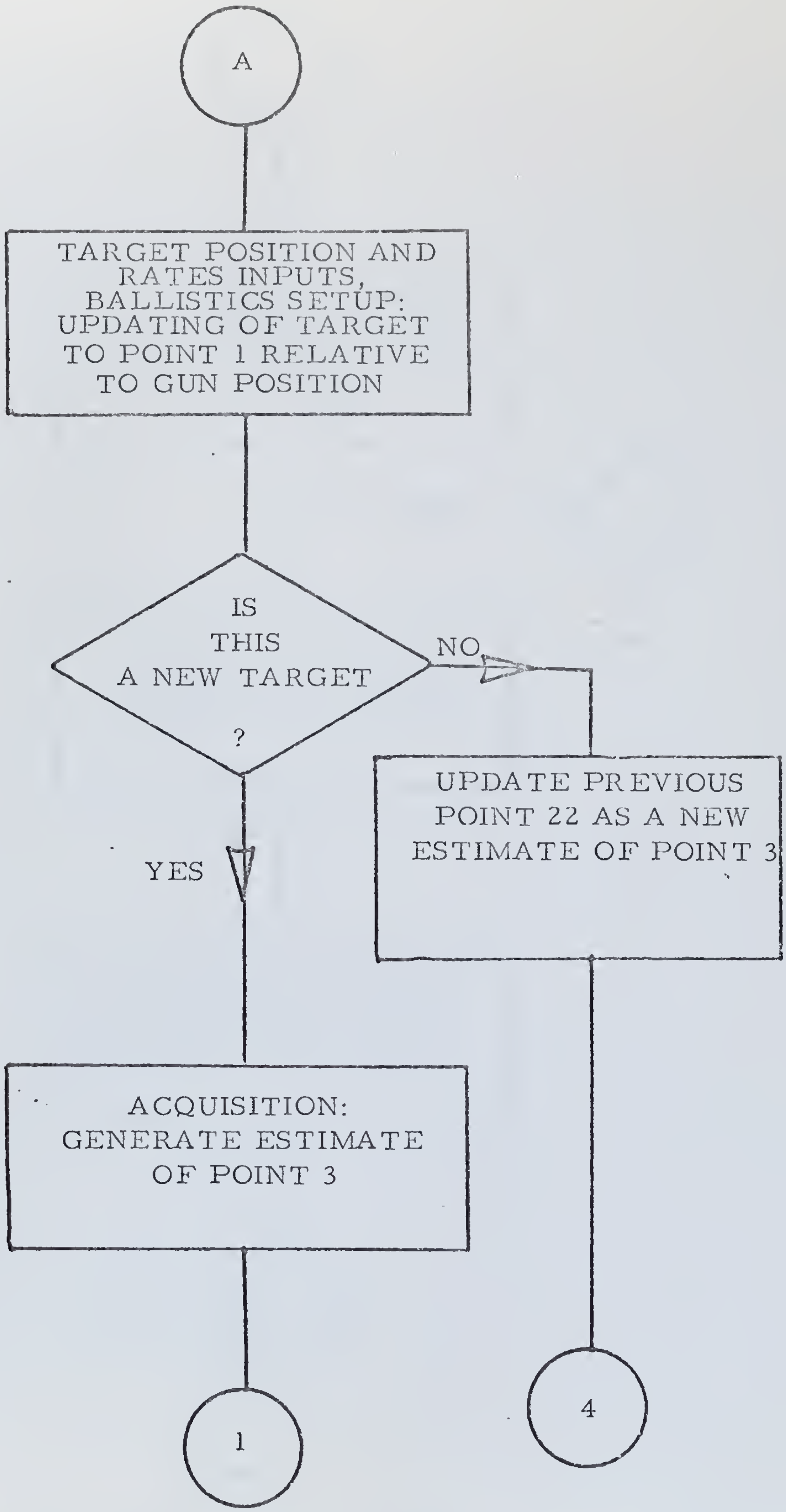

Figure 2

FUNCTIONAL FLOW CHART

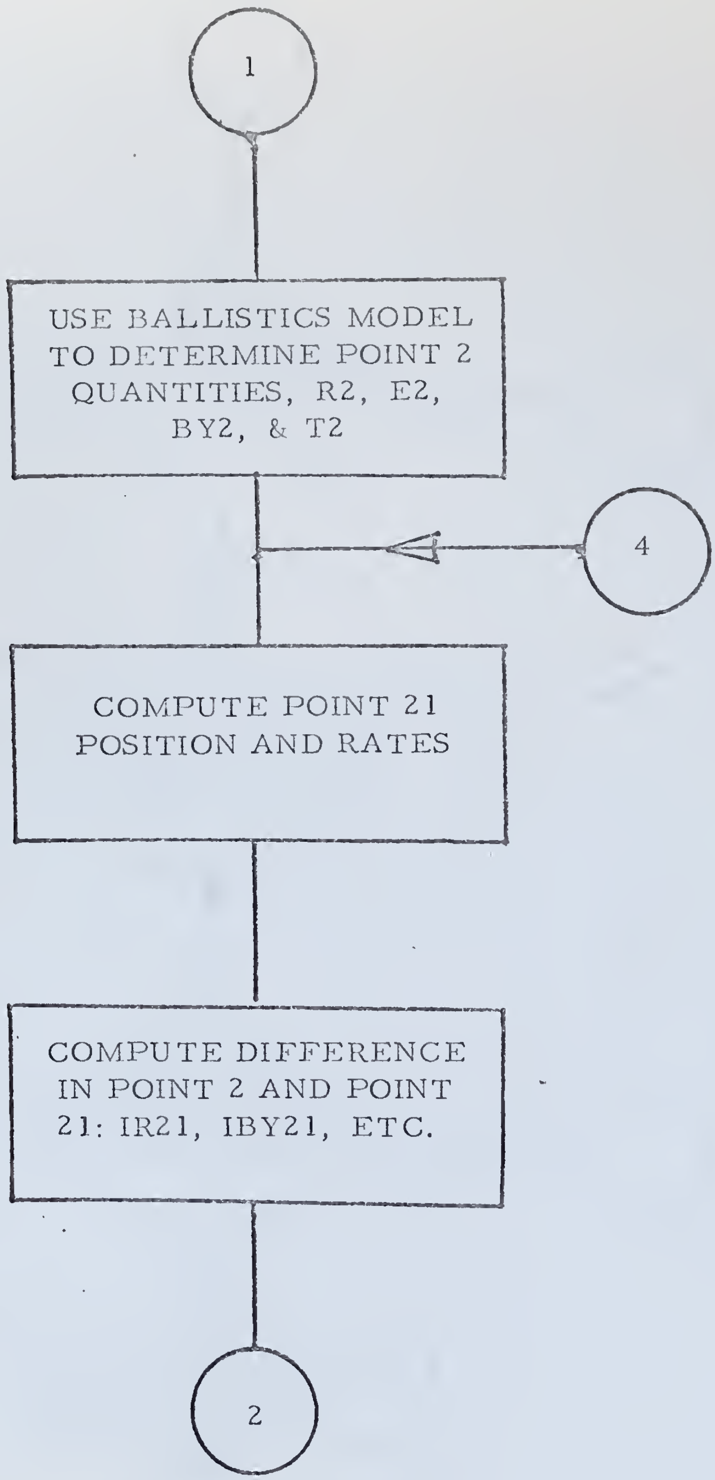

Figure 2

FUNCTIONAL FLOW CHART

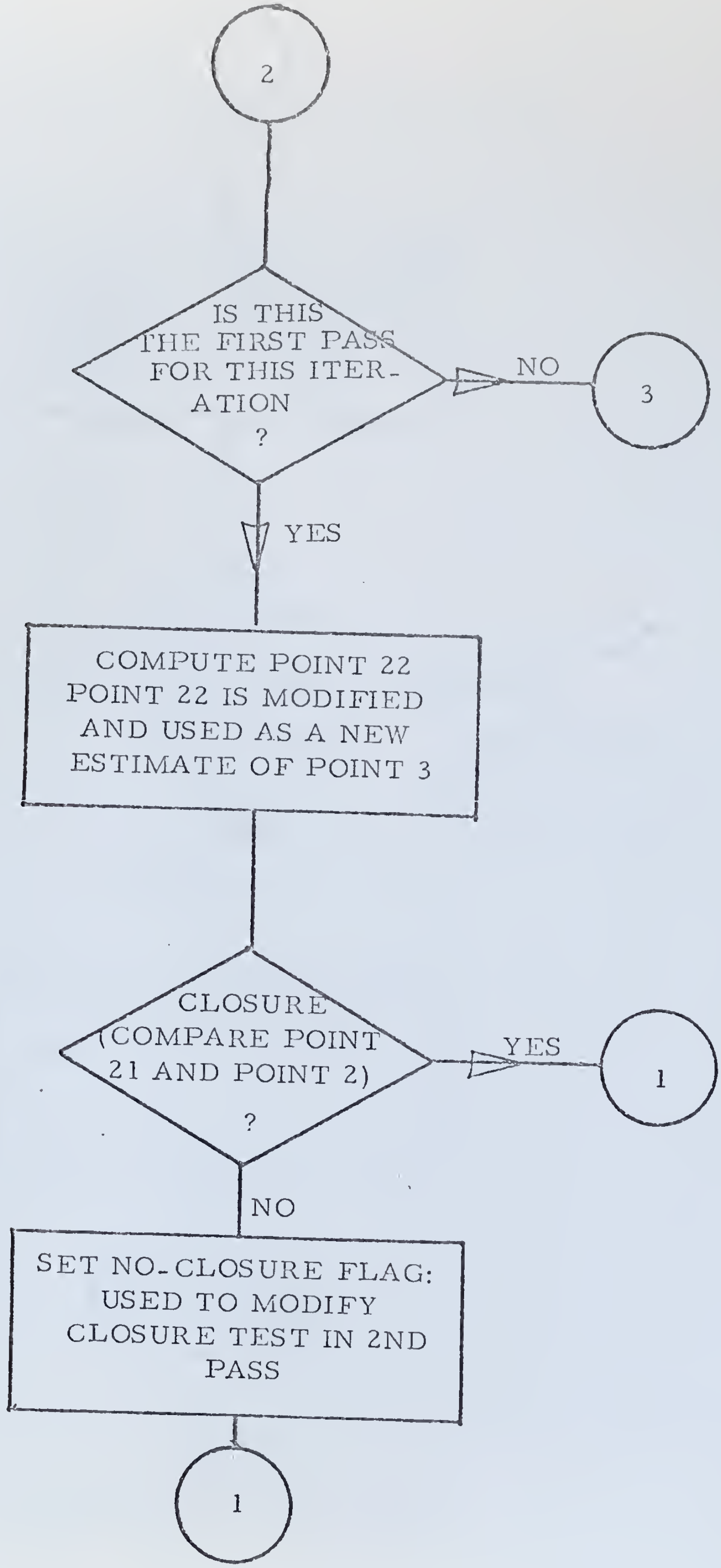

Figure 2

FUNCTIONAL FLOW CHART

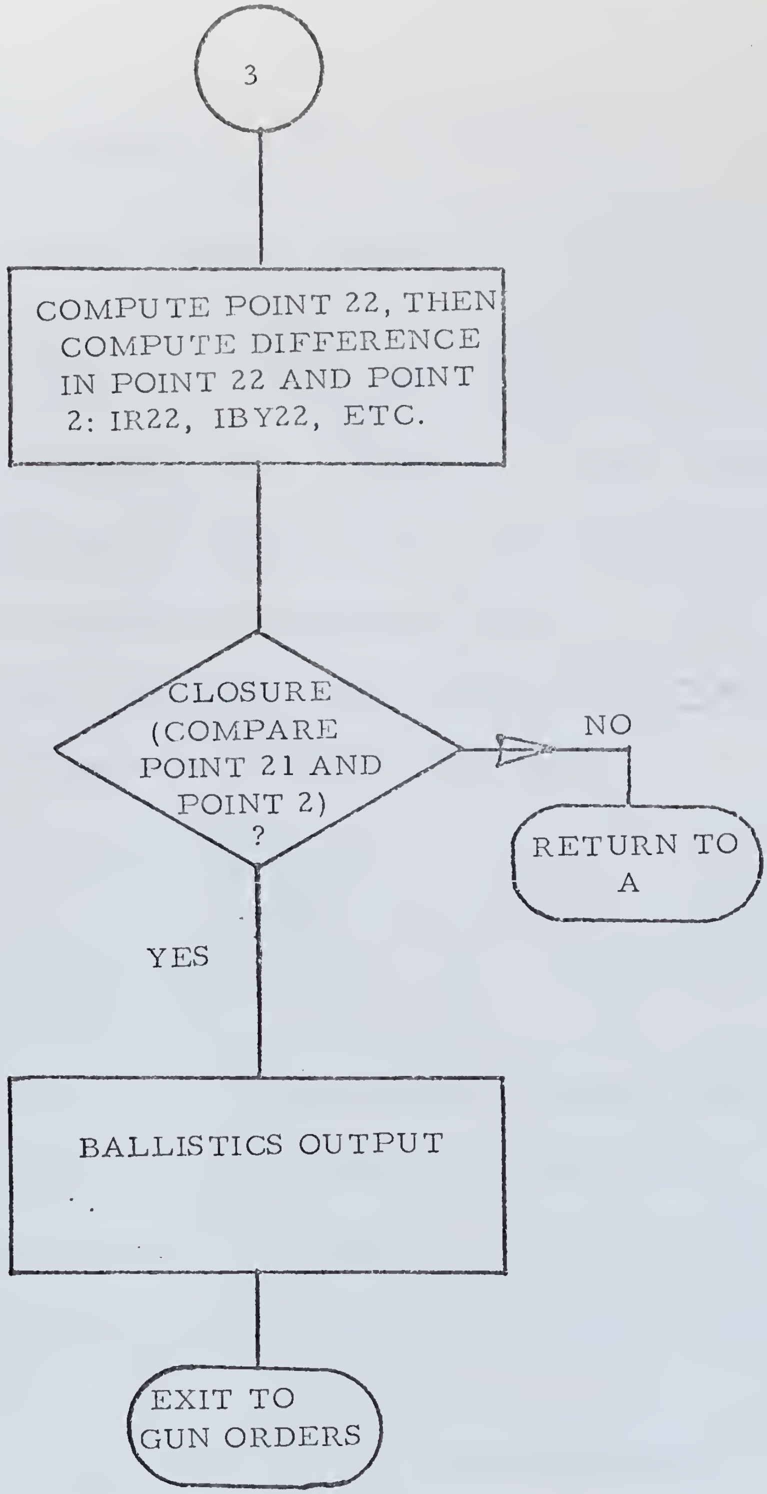

Figure 2

FUNCTIONAL FLOW CHART

III. BALLISTICS EFFECTS MODELING

The MK86 ballistics modeling program provides the means for determining the values and effects of the various ballistic parameters. The approach taken and the structure of the modeling is described here for AA cartesian ballistics (see Figure 3). The basic function of the model is to provide projectile impact or intercept point parameters based on advance target position parameters (that is, to provide point 2 output data from point 3 input data). The ballistics parameters are broken down into the three categories described below.

A. CATEGORY-ONE PARAMETERS

1. Gun elevation angle, EG
2. Time of flight, T3
3. Range effect per unit initial velocity variation, DRDU
4. Range effect per unit air density variation, DRDH
5. Range effect per unit air temperature variation, DRDTF
6. Time of flight effect per unit initial velocity variation, DT3DU
7. Time of flight effect per unit air density variation, DT3DH
8. Time of flight effect per unit air temperature variation, DT3DTF

Each of these parameters is modeled by a cubic polynominal function of range of the form:

$$F=K0+K1(IR3)+K2(IR3)^2+K3(IR3)^3$$
$$=K0+IR3(K1+IR3(K2+K3(IR3)))$$

Where F is the parameter to be determined and the K's are coefficients obtained from tables which are arranged in elevation "stacks", each of which is "partitioned" according to slant range. This is illustrated by the table below:

TABLE 1

Typical Ballistics Table for One Parameter (such as Time of Flight)

Stack 1	Stack 2	Stack N
Stack elevation angle #1	Stack elevation angle #2	Same format as other stacks
List of range intervals in this stack	List of range intervals in this stack	
Coefficients for first range interval	Coefficients for first range interval	
Coefficients for 2nd range interval	Coefficients for 2nd range interval	
etc.	etc.	

The quantity IR 3 in the above cubic polynominal is the incremental slant range (i. e. the slant range of interest minus the lower range bound of the range partition in use).

That is, each category-one parameters is modeled by a cubic equation in each range-elevation partition, as shown in Figure 4. It is the coefficients of the cubics that are provided by the ballistic tables. In the simplest case, two values of F are computed, one corresponding to each of the elevation stacks which bracket the actual elevation. The final value of F then follows from linear interpolation.

B. CATEGORY-TWO PARAMETERS

1. Coriolis effects on range, crossrange and time of flight
2. Drift effect
3. Angle of fall factor and calculation of tangent of the angle of fall

Each of these parameters is modeled by an analytic expression modified by a correction factor which is calculated in essentially the same way as a category-one parameter. In this case the factor K is computed by a linear incremental slant range relation, K=C0+C1(IR3).

C. CATEGORY-THREE PARAMETERS

1. Range effect per unit projectile weight variation
2. Time of flight effect per unit projectile weight variation
3. Wind and motion effects on range and crossrange

These parameters are calculated in special subroutines which do not require range and/or elevation dependent table lookup.

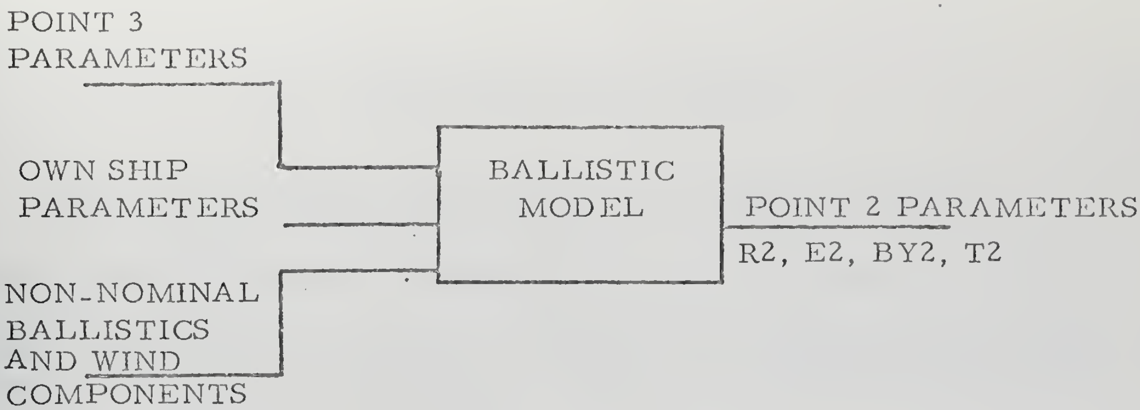

Figure 3

BALLISTIC MODEL

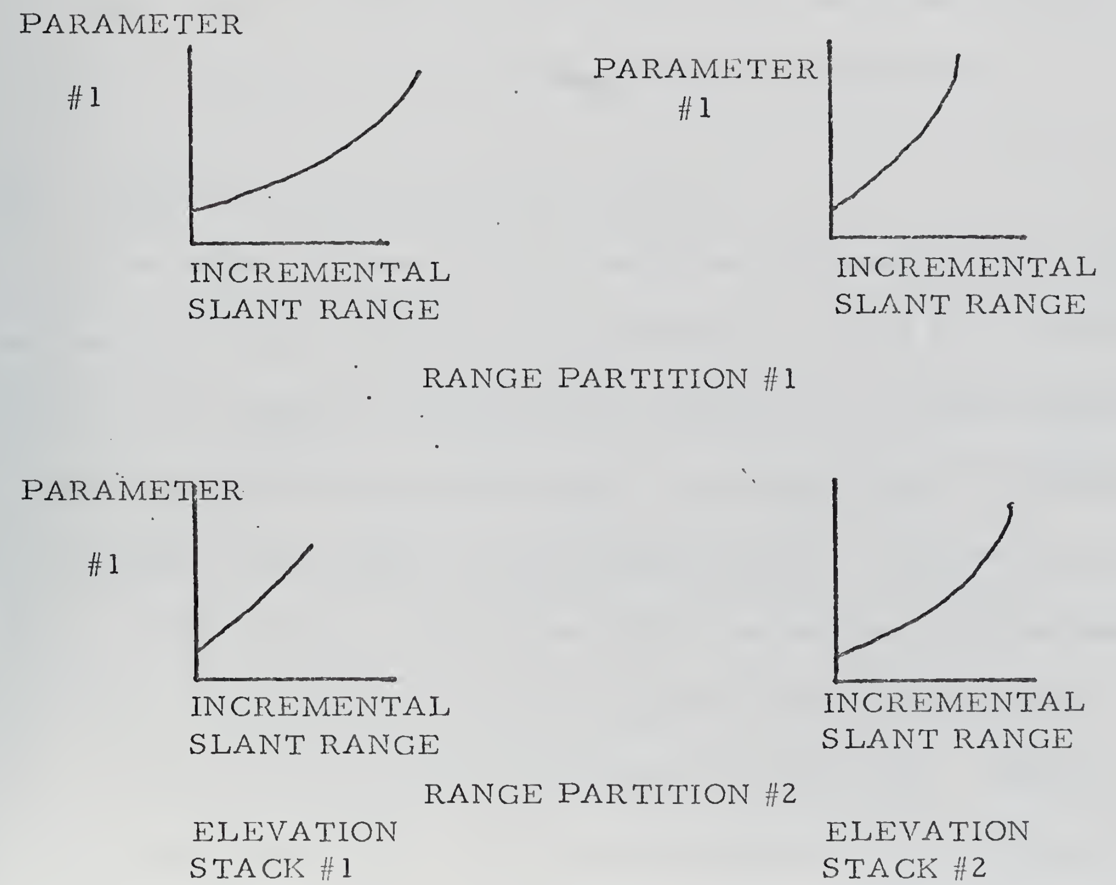

Figure 4

PARAMETER AS A FUNCTION OF ELEVATION AND RANGE

IV. THE BALLISTICS TABLES

The ballistics tables were derived by modeling range table data provided LEC by NWL Dahlgren [Ref. 1]. There are six sets of tables. Each table corresponds to one or more ballistics parameters. The AA tables are of the following form:

1. Each parameter's section of the table is divided into elevation stacks.
2. Each elevation stack is divided into range partitions or intervals.
3. For each range partition there are a set of coefficients which are used to evaluate the chosen parameter.

A. MK86 MACHINE LANGUAGE TABLES

The tables are located in the computer in sequential order. The address of the first entry for each parameter is stored in the ETABAD matrix. This address is also stored in ISRCH(1). To find the address of the first elevation stack to be used, the address stored in ISRCH(1) is modified by the contents of the PUSHTB matrix. The first 8 bits of each PUSHTB entry contains the number of entries to be bypassed in order to locate the address of the initial elevation stack for the low elevation AA mode (advance elevation less than 35°). The last 8 bits indicate the number of entries to be bypassed for the high elevation AA mode. The modified address is then stored in ISRCH(2).

The first entry of each elevation stack contains:

1. Stack elevation angle

2. Number of range partitions, minus one

3. Internal shift count

The stack elevation angle is stored in STACK(3), the number of range partitions is stored in the NP matrix, and the internal shift count necessary to position the binary point in the KSH matrix.

A complete listing of the tables may be found in reference 2.

B. BALLISTICS TABLES IN THE FORTRAN SIMULATION

The entry into the the organization of the tables is different in the FORTRAN simulation program from that in the machine language program.

The ballistics tables are stored in a three-dimensional array. The coordinates of the matrix are parameter, elevation stack angle, and the location of a particular item in the stack. The first entry in a particular elevation stack is the elevation stack angle, and it is followed by the number of range partitions. Then the limits of the range partitions are listed, followed by the coefficients for each range partition. The location in a stack of the first coefficient required is calculated by multiplying the number of coefficients times the number of the range partition of interest and adding the location of the first range partition plus 3. Then the location is stored in T17. A listing of the ballistics tables starts on page 90 . In this listing,

each table corresponds to a ballistic parameter (see Table 2), and each column corresponds to an elevation stack.

TABLE 2

Ballistic Tables (FORTRAN)

Table	Parameter
1	Time of flight (T3)
2	Super-elevation angle (EGB)
3	Range effect due to change in initial velocity (DRDU)
4	Time of flight effects due to change in initial velocity (DT3DU)
5	Range effect due to change in air density (DRDH)
6	Time of flight effects due to change in air density (DT3DH)
7	Range effect due to change in air temperature (DRDTF)
8	Time of flight effects due to change in air temperature (DT3DTF)
9	Tangent of angle of fall (TANW)
10	Coriolis effects
11	Drift effects

V. BALLISTICS SUBPROGRAMS

Since this project was concerned only with the AA aspect of the GFCS MK86 system, all references to surface ballistics, grid mode, ROS mode, TDT mode, and spherical ballistics have been omitted. For the flow chart of the subprograms see Figure 5. A complete explanation and flow chart may be found in References 3 and 4.

A. TARG1

This subprogram freezes or holds the console inputs for the duration of the ½ second ballistics computation cycle, and calculates and sets the following indicators:

1. AA ballistics
2. Cartesian ballistics
3. Ammo type, charge
4. Gun and gun control console assignments
5. Target assignment
6. Acquisition or tracking ballistics

In addition, TARG1 computes non-nominal increments that may exist in various ballistics parameters, performs a ship's course update, and determines gun parallax corrections. Calculations are also performed to update the target data to "ballistic time" (i. e., the next "integer"¼ second).

B. TARG2

This subprogram performs the acquisition ballistics computations for AA cartesian ballistics. Its function is to calculate point 3 (the theoretical aiming point) from point 1 (the target position at ballistics time). This subprogram determines the first advance position (aim point) when no valid prior advance position exists. This situation would occur whenever:

1. A new AA target has been assigned
2. There was no ballistics solution during the previous ½-second ballistics computation cycle
3. There has been a change in the mode assignment for the gun
4. There has been a change in the ammo type or charge

The countdown flag (KGOOD) is set to one, indicating that the acquisition subprogram must be completed more than once before the gun is released from any previous state it may have been in. The acquire flag is set to one to indicate the acquisition ballistics mode.

The advanced elevation (E3) is set equal to the elevation angle at point 1 (E1). TMONIT, which controls the ballistic table computations, is then entered to find the nominal time of flight (T1) from RC, the range to point 1.

C. BCOMPA

This subprogram computes the time of flight to point 3 (T3) and its derivatives with respect to range and elevation. It also computes the mixed second partial derivative with respect to range and elevation. Using E3 and R3 as inputs, TMONIT is called and the time of flight is returned in TEMP2, the range derivative (DT3DR) in TMP3, the elevation derivative (DE3DE) in DPDEP, and the mixed partial (DT3RDE) in DPDED.

D. BCOMPB

This subprogram computes the super elevation angle (EGB) and its derivatives with respect to range and elevation, as well as the mixed second partial with respect to range and elevation. TMONIT is called and the super elevation angle is returned in TEMP2, the range derivative in TMP3, the elevation derivative (DEGDE) in DPDEP, and the mixed partial (DEGRDE) in DPDED. Then EGB and E3 are summed to form the total elevation angle E4=E3+EGB, which is limited to a maximum of 85 degrees.

E. BCOMPC

This subprogram computes the derivative of range with respect to a non-nominal increment of projectile velocity (DRDU) and the derivative of DRDU with respect to elevation (DRDUDE). Then the actual range effect (URH) due to non-nominal projectile velocity is computed.

F. BCOMPD

This subprogram computes the derivative of the time of flight with respect to the increment in projectile velocity (DR3DU) and the time of flight effect (UT3) due to non-nominal projectile velocity. The mixed second partials of time of flight with respect to increment in projectile velocity and elevation (DT3UDE) and range (DT3UDR) are also computed.

G. BCOMPE

This subprogram computes the derivative of range with respect to non-nominal increments in air density (DRDH) and the derivative of DRDH with respect to elevation (DRDHDE). Then the actual range effect due to non-nominal ballistic air density (HRH) is computed.

H. BCOMPF

This subprogram computes the derivative of time of flight with respect to a non-nominal increment in air density (DT3DH) and the actual time of flight effect (HT3) due to DT3DH. The mixed second partials of time of flight with respect to air density and range and elevation (DT3HDR, DT3HDE) are also computed.

I. BCOMPG

This subprogram computes the derivative of the range with respect to a non-nominal increment in surface air temperature (DRDTF),

the actual range effect (TFR3), and the mixed partial of range with respect to temperature and elevation (DRTFDE).

J. BCOMPH

This subprogram computes the derivative of time of flight with respect to a non-nominal increment in surface temperature (DT3DTF); computes the actual time of flight effect (TFT3) due to air temperature; and calls the weight effect subprogram (RTWT).

The subprogram then performs a partial sum of all the range effects and time of flight effects calculated so far and calculates the derivative of the time of flight to point 2 (the predicted impact point) with respect to range (DT2DR).

K. BCOMPI

This subprogram computes the downrange and crossrange components of gun motion (DMR3OG, DMGOG) before calling GUNMOT, the gun motion subprogram. It then computes the tangent of the angle of fall (TAN W) and calls the wind effects subprogram, WIND.

L. BCOMPJ

This subprogram obtains a first partial sum of range effects, a second partial sum of time of flight effects, and calls CORLIS to obtain coriolis effects.

M. BCOMPK

This subprogram computes the intercept geometry by determining the horizontal range, vertical range, slant range, and bearing to the predicted projectile impact point (point 2). It also computes the elevation angles for the gun.

N. TARG3

This subprogram performs the iteration and closure computations. Its function is to update through two iterations the previous advanced aim point (point 3) such that satisfactory coincidence is obtained between the predicted impact point (point 2) and the predicted target position at impact time (point 21). The closure equations determine when there is sufficient closeness between the two points for an acceptable solution.

The input values to TARG3 come from two sources. In the tracking mode, the data from the previous pass through TARG3 to BCOMPK is used to update current values. In the acquisition mode, the values calculated in subprograms TARG2 through BCOMPK are used.

O. TARG4

This subprogram computes the ballistics outputs. It computes the final inertial gun orders, gun order rates, and fuse time rate as well as the inertial impact coordinates and rates for eventual display

on the B-scan. These computations are oriented to the start of the next 1/16 second time-slice.

P. TMONIT

For a given parameter, this subprogram does the initial setup and locates the proper table and elevation stack corresponding to the target's elevation. Then it calls appropriate subprograms to calculate the value of the parameter from the chosen table.

In more detail, either two or three stacks are selected so that their elevation values bracket the value of E3. Two are selected for parameters which are linearly interpolated with respect to elevation and three for those which are parabolicaly interpolated. The partition selection, coefficient look-up, and polynominal evaluation with respect to range are performed on each of the selected sets of coefficients yielding a value of the parameter and its derivatives corresponding to each of the bracketing elevations. The final value of the parameter and its derivatives are then obtained by interpolating with respect to elevation.

Q. AASRCH

This subprogram controls the selection of coefficients for bracketing elevation stacks, the cubic polynominal evaluation, and the interpolation between elevation stacks. It also controls the calculation of the appropriate derivatives. For standard tables (four

coefficients per partition) the address of the first coefficient is found by calling the SEARCH subprogram. Then the polynomial subprogram is called to evaluate the parameter, which is stored, with its derivative, in the STACK matrix. When all parameters and derivatives corresponding to the selected elevation stacks have been evaluated, then the interpolation subprogram is called to get the final values. For non-standard tables, only the address of the first coefficient is found for each selected elevation stack. These values are stored in the HOLDB3 matrix.

R. SEARCH

This subprogram, which is called by AASRCH, finds the correct range partition and the location of the first coefficient for that range partition.

S. POLYN

This subprogram, which is called by AASRCH, evaluates the polynomial from the coefficients and finds the derivatives with respect to range for each elevation stack chosen.

T. INTERP

This subprogram performs linear or parabolic interpolation on a set of two or three parameter values with respect to elevation. The interpolates value is the value of the parameter at the actual advance elevation.

U. DECSI

This subprogram decodes the elevation information from each stack and stores it in STACK(3).

V. SLIDE

This subprogram moves the parameters stored in the STACK matrix up one location.

W. FINDAL

This subprogram finds the location of the next available elevation stack.

X. CORLIS

This subprogram computes ballistic Coriolis effects for slant range, altitude, time of flight, crossrange and downrange effects. The Coriolis table is a non-standard table with six coefficients per range partition. Three linear coefficients are computed and used to determine the effects of Coriolis.

Y. DRIFT

This subprogram computes the crossrange effect of projectile drift. The drift effect table is a non-standard table with two coefficients per range partition. Two polynomials are evaluated for AA ballistics, one for each elevation stack bracketing the actual advance elevation.

Z. GUNMOT

This subprogram calculates the ballistic effect due to gun motion (i.e. the advance range, time of flight, deflection and cross-range effects). Under conditions of no gun motion, the projectile leaves the muzzle with an initial velocity vector (U) at inertial true bearing BGY, and inertial elevation, EG. With gun motion, the component along the U vector will increase the magnitude of U.

AA. RTWT

This subprogram calculates the (minor) effect of projectile weight on time of flight and range. Due to inertia, an increase of projectile weight will produce a decrease of initial velocity. It will also produce a decrease in drag acceleration, which is equivalent to a decrease in air density. Thus the range and time of flight effects due to a projectile weight variation can be computed as the sum of the equivalent effects of increments in initial velocity and ballistics air density.

AB. WIND

This subprogram calculates the total effects of wind on advance range, time of flight, and crossrange.

AC. ZLIMIT

This subprogram limits the Z coordinate of a target to a minimum value, based upon target horizontal range, earth's curvature, and height of the gun above the sea surface.

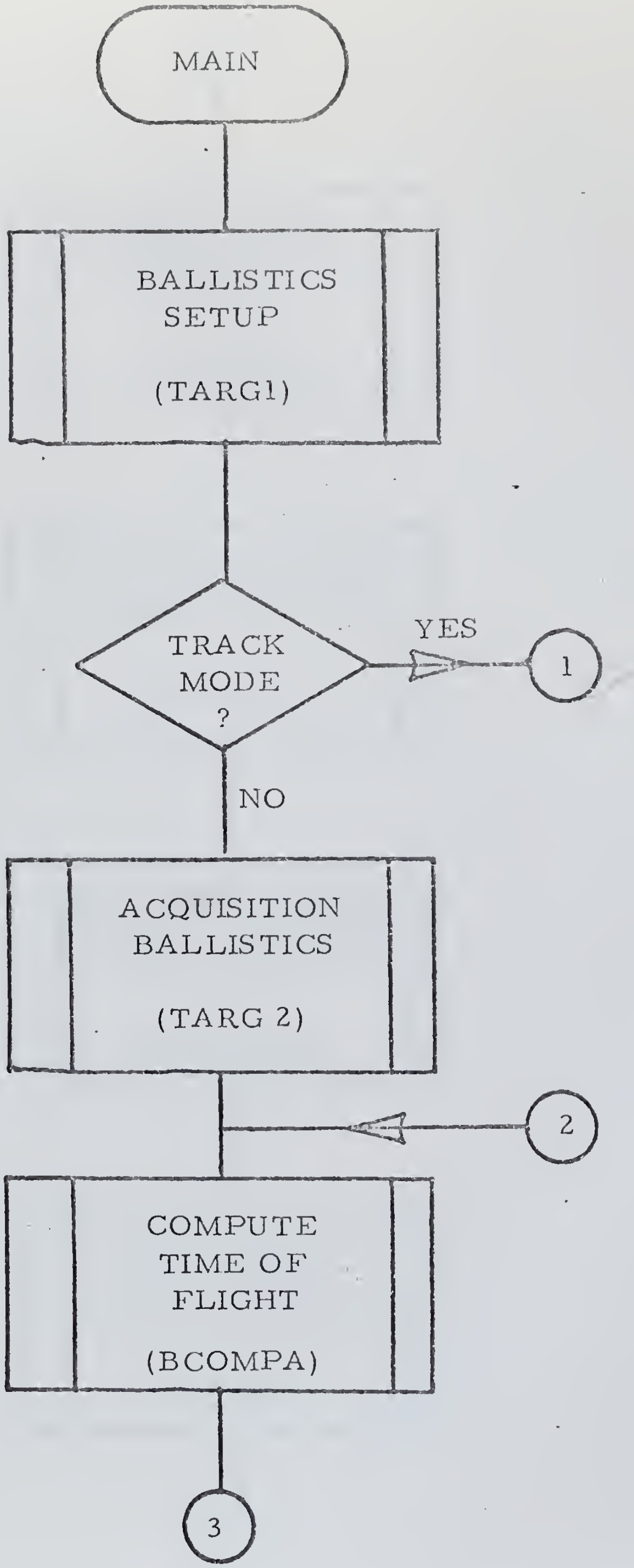

Figure 5

FORTRAN FLOW CHART

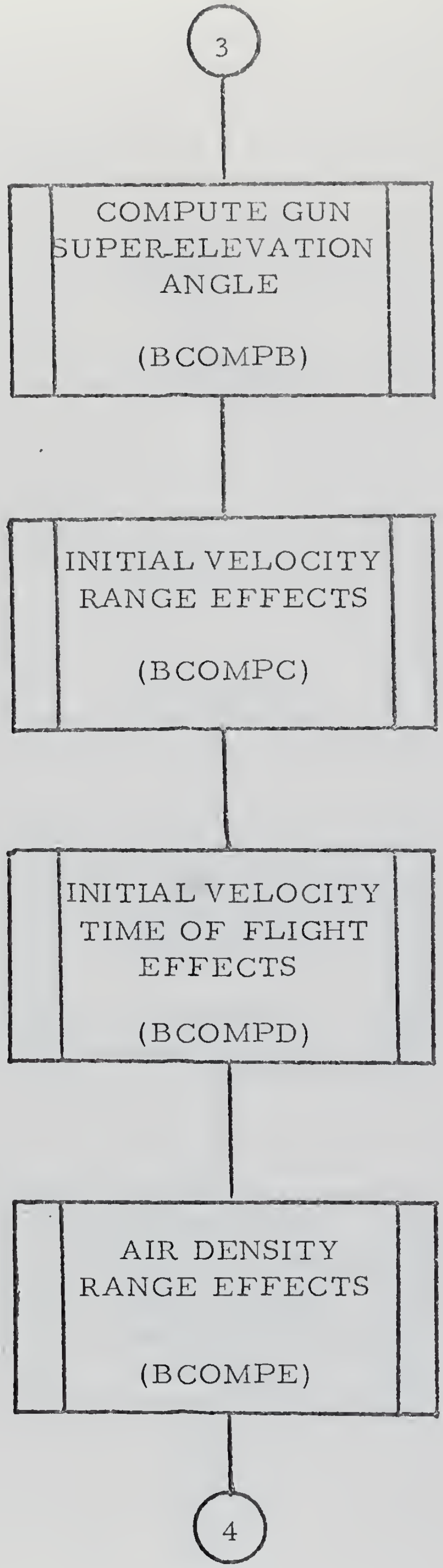

Figure 5

FORTRAN FLOW CHART

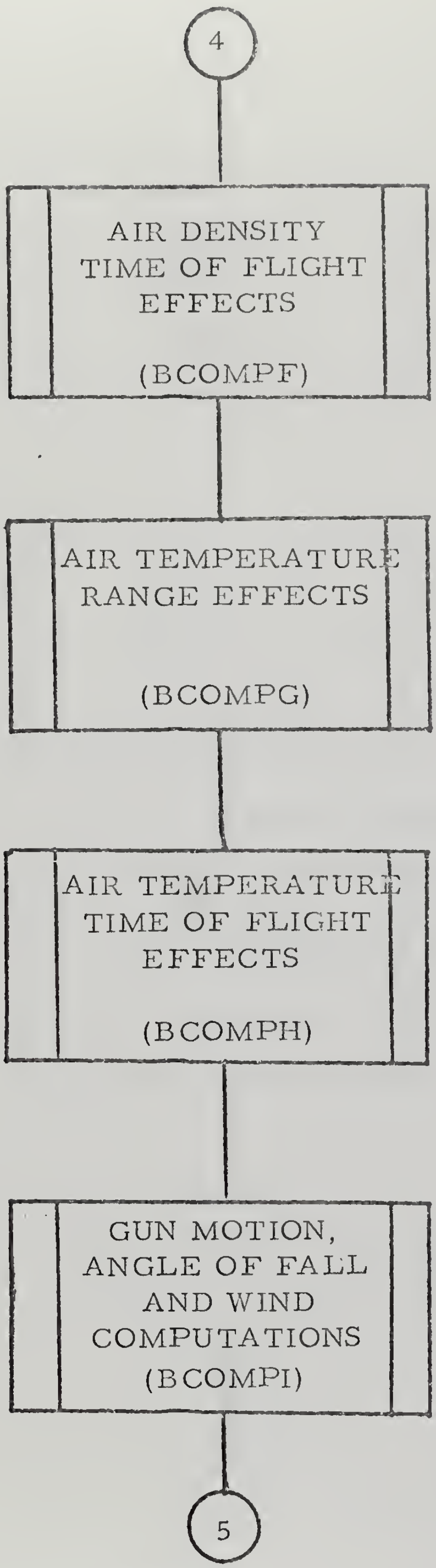

Figure 5

FORTRAN FLOW CHART

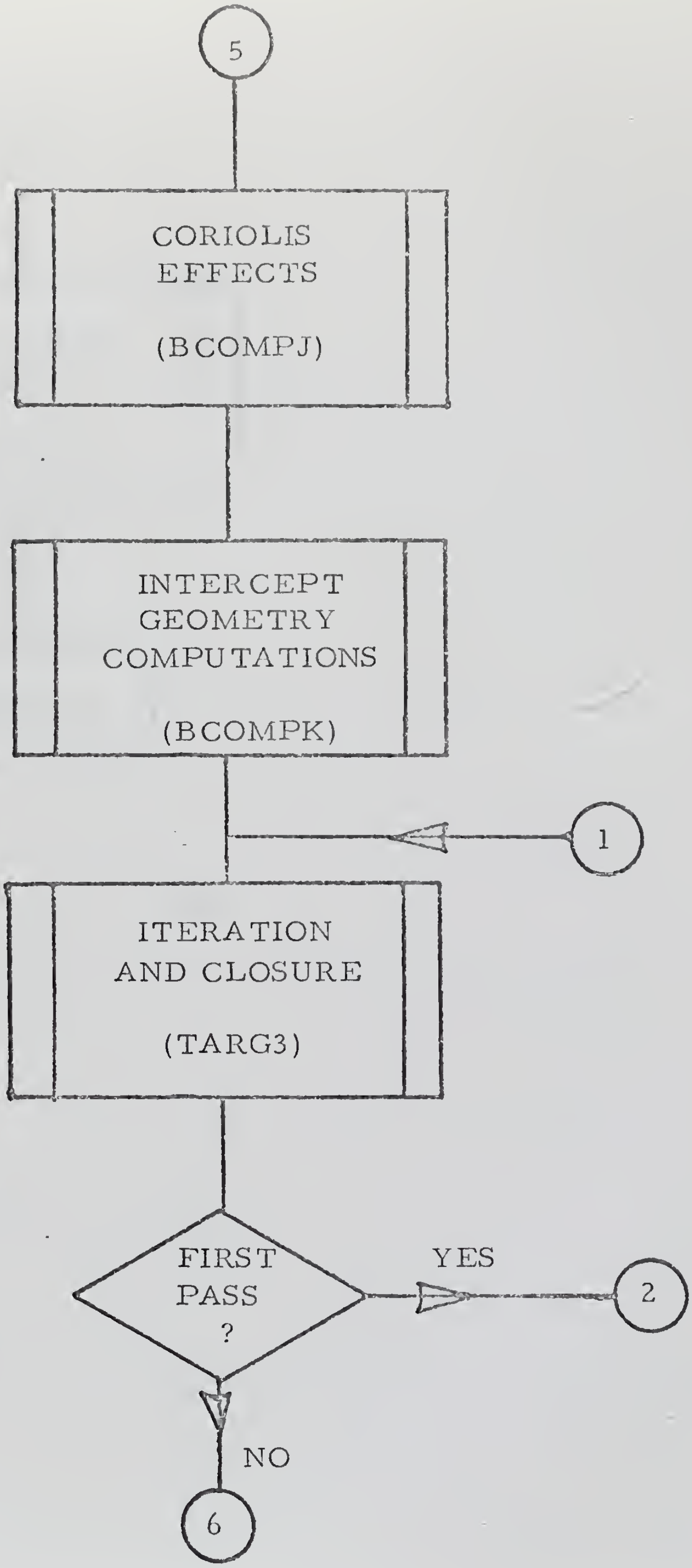

Figure 5
FORTRAN FLOW CHART

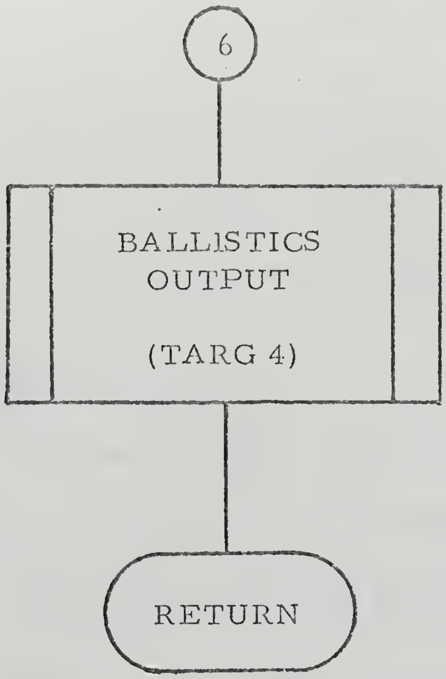

Figure 5
FORTRAN FLOW CHART

VI. FORTRAN SIMULATION

Since only the AA mode was to be considered, all flags and variables necessary to indicate the AA mode were pre-set to insure that only the AA cartesian ballistics computations were performed.

A. ASSUMPTIONS MADE

The FORTRAN program has no real time clock, therefore certain timing assumptions had to be made. These assumptions are:

1. JOELCK (target valid data-time for ballistics) equals -1/32 second
2. TBR (saved value of ballistic time) was initialized to 0
3. TB (ballistics time to next second) initialized to ¼ second
4. TBFIN (final ballistics time for next 1/16 second initialized to 15/32 second

B. FORTRAN PROGRAM

The program, which is listed starting on page closely follows the arrangement of the MK86 machine language program. It has been suitably annotated with comment cards, and otherwise should be self-explanatory. It has been written so that all ranges are in yards, and all angles are in degrees.

Three internal subprograms (DECSI, SLIDE, AND FINDAL) of TMONIT in the machine language program have been treated as separate subprograms in the FORTRAN program. Also, in BCOMPB the super-elevation angle is computed in minutes and then converted to degrees.

C. VERIFICATION

Complete verification of the FORTRAN program was not done due to lack of sufficient data. The separate subprograms were verified by calculations done by hand. Using data furnished by LEC, a partial verification was attempted. However, effects such as wind, coriolis, non-nominal parameters and ship's motion had been set to zero in the data provided [Ref. 5].

```
      COMMON/RT/COS1,SIN1,TAN
      COMMON/CA/T3,DT3DR,DT3DE,DT3RDE
      COMMON/CB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4,DEGRDE
      COMMON/CC/DRDU,DRDUDE,URH
      COMMON/CD/DT3DU,DT3UDR,DT3UDE,UT3
      COMMON/CE/DRDH,DRDHDE,HRH
      COMMON/OF/HT3,DT3DH,DT3HDE,DT3HDR
      COMMON/CG/DRTFDE,DRDTF,TFR3
      COMMON/OH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
      COMMON/ I/ANGLEW,DMR3OG,DMBOG,SINEGB,COSEGB,TANEG,
     CTANW,TOVERW
      COMMON/CJ/J2R3,R2J2,T2J2
      COMMON/CK/R2,E2,T2,BY2
      COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
      COMMON/T01/RCOEFF(8),KMTABL(11,2),ETABL1(17,11),PUSHTB(11,2)
      COMMON/CGU/DRDMEY,DRDMRY,DT3DMY,DT3MEY,MR3CR3,MR3OT3,MOLS,MORLS
      COMMON/CDR/DORLS
      COMMON/CCR/KRLS,KR3,KTRRE3,KT3,T2J2SQ,TANE4
      COMMON/CW/DRDWRH,DTDWRH,WBG,WRHG,WRLS,WR3,WT3
      COMMON/OZ/RHGEN,RZGEN
      COMMON/N1/ICODEP,DPDEP,DPDEC,TEMP2,TMP3,TMP10
      COMMON/N2/BCCUNT
      COMMON/T1/ROX,ROY,DMOX,DMOY,ROXY,ROYY,RXL,RYL,RZL,DMCXY,DMOYY,
     CDRTX,DRTY,DRTZ
      COMMON/T12/SINLAT,COSLAT,CQO,DCQC,SINLAR,COSLAR
      COMMON/T13/WX,WY,WXR,WYR
      COMMON/T24/QALS,QAR2,QAV,ERVG,QR3,QRLR,QVR,RV
      COMMON/T151/ZSUBN(3),CDMRTZ(3)
      COMMON/T16/DSU,DTF,DWT,PWDTMP,DH
      COMMON/T17/BALL1,NEW3DT,NOCLOS,AQUIRE,ICCUNT,KGOOD,BALIND
      COMMON/T18/AARO,RNGDOT,RC
      COMMON/T19/TB,TBR
      COMMON/T120/IH,IPT,ITF,IU,IWT,HO,UO
      COMMON/T121/E1,BYL
      COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/T3/RX21,RY21,DMRT21,DE21,DBY21,E21
      COMMON/T31/BY22,IBY22,E22,R22,IT2
      COMMON/T4/BGYAUP,BTT2,DBGY,DEG,DT216,DT5,EGAUP,FLASH,LITSET,GE,QLS,TT2
      INTEGER*4 T17,PUSHTB
      REAL*4 LCQO
      REAL*4 IR3
      REAL*4 ITB
      REAL * 4 KRLS,KR3,KTRRE3,KT3
      REAL*4 IT2,IT2X
```

```
      INTEGER*4 ETABL1
      REAL*4 JOELCK
      DO 1 I=1,8                                                        MAI00050
      READ(5,10) RCOEFF(I)                                              MAI00060
   10 FORMAT (E15.8)                                                    MAI00070
    1 CONTINUE                                                          MAI00080
      DO 2 I=1,17
      READ (5,11) (ETABL1(I,J),J=1,11)                                  MAI00100
   11 FORMAT (11I5)                                                     MAI00110
    2 CONTINUE                                                          MAI00120
      DO 3 I=1,11                                                       MAI00130
      STACK(I)=0                                                        MAI00140
    3 CONTINUE                                                          MAI00150
      DO 4 I=1,11                                                       MAI00160
      READ (5,12)(KMTABL(I,J),J=1,2)                                    MAI00170
   12 FORMAT(2I5)                                                       MAI00180
    4 CONTINUE                                                          MAI00190
      DO 5 I=1,11                                                       MAI00200
      READ(5,12) (PUSHTB(I,J),J=1,2)                                    MAI00210
    5 CONTINUE                                                          MAI00220
      DO 6 J=1,15
      DO 7 N=1,32                                                       MAI00240
      READ(5,13) TABLE(1,J,N)                                           MAI00250
   13 FORMAT (E15.8)                                                    MAI00260
    7 CONTINUE                                                          MAI00270
    6 CONTINUE                                                          MAI00280
      DO 8 J=1,17
      DO 9 N=1,32
      READ(5,13)TABLE(2,J,N)
    9 CONTINUE
    8 CONTINUE
      DO  20 J=1,11
      DO 21 N=1,12
      READ(5,13) TABLE(3,J,N)
   21 CONTINUE
   20 CONTINUE
      DO 22 J=1,4
      DO 23 N=1,12
      READ(5,13)TABLE(4,J,N)
   23 CONTINUE
   22 CONTINUE
      DO 24 J=1,8
      DO 25 N=1,17
      READ(5,13) TABLE(5,J,N)
   25 CONTINUE
   24 CONTINUE
      DO 26 J=1,4
      DO 27 N=1,12
```

```
      READ(5,13) TABLE(6,J,N)
   27 CONTINUE
   26 CONTINUE
      DO 28 J=1,11
      DC 29 N=1,17
      READ(5,13) TABLE(7,J,N)
   29 CONTINUE
   28 CONTINUE
      DC 31 J=1,4
      DO 32 N=1,12
      READ (5,13) TABLE(8,J,N)
  32  CONTINUE
   31  CONTINUE
      DC 33 J=1,4
      DC 34 N=1,17
      READ (5,13) TABLE(9,J,N)
   34 CONTINUE
   33 CONTINUE
      DO 35 J=1,3
      DC 36 N=1,9
      READ (5,13) TABLE(10,J,N)
   36 CONTINUE
   35 CONTINUE
      DC 37 J=1,5
      DC 38 N=1,11
      READ(5,13) TABLE(11,J,N)
   38 CONTINUE
   37 CONTINUE
       KCRIND=4
      RHOLD=0                                                           MAI00300
      RTEMP=0                                                           MAI00310
      TEMP2=0                                                           MAI00320
      TMP3=0                                                            MAI00330
      TMP10=0                                                           MAI00340
      ICODEP=0                                                          MAI00350
      DPDEP=0                                                           MAI00360
      DPDED=0                                                           MAI00370
      BCOUNT=0                                                          MAI00380
      K=0                                                               MAI00390
C     TEST RUN VALUES ALL SHIP'S PARAMETERS SET TO 0, USE ALL NONIMAL VALUES
C     FOR AMMO
C
      PWDTMP=90
      DSU=2500
      DTF=59
      DH=100
      WX=0
      WY=0
```

```
      QALS=0
      QAR2=0
      QAV=0
      DCQO=0
      RCX=0
      RCY=0
      DMOY=0
      DMOXY=0
      CGC=0
      BALL1=1
      RNGDOT=-280
      AARC=2300
      ERVG=975
      COSLAT=1.0
      SINLAT=0.0
      DO 1000 J=1,3
      DO 200 N=1,3
      READ(5,13) CDMRTZ(N)
      WRITE(6,700) CDMRTZ(N)
700   FORMAT(10X,E15.8)
200   CONTINUE
      DO 201 N=1,3
      READ(5,13) ZSUBN(N)
      WRITE(6,700) ZSUBN(N)
201   CONTINUE
      CALL TARG1
      IF(BCOUNT.EQ.0) GO TO 1000
      IF (BCOUNT.GT.1) GO TO 301
      CALL TARG2
300   CALL BCOMPA
      CALL BCOMPB
      CALL BCOMPC
      CALL BCOMPD
      CALL BCOMPE
      CALL BCOMPF
      CALL BCOMPG
      CALL BCOMPH
      CALL BCOMPI
      CALL BCOMPJ
      CALL BCOMPK
301   CALL TARG3
      IF (BCOUNT.LT.14) GO TO 300
      CALL TARG4
1000  CONTINUE
      STOP                                                              MAI00440
      END                                                               MAI00450
```

```
      SLBROUTINE TARG1                                                  TAR00010
C                                                                       TAR00020
C     THIS SUBTCUTIME FREEZES OR HOLD CONSOLE IMPUTS FOR 1/2 SECOND     TAR00030
C     AND CALCULATES OR DETERMINES THE FOLLOWING INDICATORS;AA          TAR00040
C     BALLISTICS,CARTESIAN BALLISTICS,AMMOTYPE,CHARGE,GUN AND           TAR00050
C     GLN CONTROL CONSOLE ASSIGNMENTS,TARGET ASSIGNMENT,                TAR00060
C     BALLISTICS MODE, ACQUISITION OR TRACKING BALLISTICS. IN           TAR00070
C     ADDITION,TARG1 COMPUTES NON-NOMINAL INCREMENTS THAT               TAR00080
C     MAY EXIST IN VARIOUS BALLISTICS PARAMETERS, PERFORMS A SHIP'S     TAR00090
C     CCURSE UPDATE, AND DETERMINES GUN PARALLAX CORRECTICNS.           TAR00100
C     CALCULATIONS ARE ALSO PERFORMED TO UPDATE THE TARGET DATA FOR THE TAR00110
C     VARIOUS BALLISTICS MODES TO BALLISTICS TIME ,NEXT INTEGER 1/4     TAR00120
C     SECOND                                                            TAR00130
C                                                                       TAR00140
      COMMON/RT/COSL,SINL,TAN
      COMMON/N2/BCOUNT
      COMMON/T121/E1,BYL
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UO
      COMMON/T19/TB,TBR
      COMMON/T18/AARO,RNGDOT,RC
      COMMON/T17/BALL1,NEW3DT,NOCLOS,AQUIRE,ICCLNT,KGOOD,BALINC
      COMMON/T16/DSU,DTF,DWT,PWDTMP,DH
      COMMON/T151/ZSUBN(3),CDMRTZ(3)
      COMMON/T24/QALS,QAR2,QAV,ERVG,QR3,QRLR,QVR,RV
      COMMON/T13/WX,WY,WXR,WYR
      COMMON/T12/SINLAT,COSLAT,CQO,DCQO,SINLAR,CCSLAR
      COMMON/T1/ROX,ROY,DMCX,DMCY,ROXY,RCYY,RXL,RYL,RZL,DMCXY,CMOYY,
     CDRTX,DRTY,DRTZ
      COMMON/OZ/RHGEN,RZGEN
C
      REAL*4 LCQO
      REAL*4 ITB
      REAL*4 KPT
      REAL*4 JOELCK
C
      GUNASS=1
C
C     INDICATE AA MODE                                                  TAR00350
C
      GUNIBI=7
      GUNIR=GUNASS
      GLNIBR=GUNIBI
C
C     COMPUTE CHANGE IN PROJECTILE VELOCITY DUE TC CHANGE INPOWDER      TAR00390
C     TEMPATURE                                                         TAR00400
C
      IPT=PWDTMP-90
C
```

```
C     SAVE SINE AND COSINE OF SHIP'S LATITUDE FOR 1/2 SECOND            TAR00420
C
      SINLAR=SINLAT
      CCSLAR=COSLAT
C
C     SAVE TARGET'S HEIGHT FOR 1/2 SECOND                               TAR00450
C
      RV=ERVG
C
C     INDICATE LOW ANGLE FIRE                                           TAR00470
      HABIR=0
C
C     INDICATE STANDARD CHARGE                                          TAR00490
C
      STNDIR=0
C
C     INDICATE CVT FUZE                                                 TAR00510
C
C     INDICATE CVT FUZE                                                 TAR00300
      CVTMT=0
C
C     INDICATES VT AMMO                                                 TAR00530
C
      AMMOR=3
      PFIND=0
      GRID=0
      NCCLOS=0
C
C     CHECK FOR INCOMING OR OUTBOUND TARGET                             TAR00580
C     IF OUTBOUND AND RANGE GREATER THAN 50K YARDS NO SOLUTION          TAR00590
C
      IF(RNGDOT.LT.0)GO TO 1
      IF(AARO-50000.GT.0) GO TO 2
C
C     IF INCOMING AND RANGE GREATER THAN AL NO SOLUTION                 TAR00620
C
    1 AL=32000-65*RNGDOT*21.85
      RNGI=AARO-AL
      IF(RNGI.GT.0.0) GO TO 2
      MCDEM=3
C
C     COMPUTE DIFFERENCE IN AIR TEMPERATURE                             TAR00670
C
      ITF=DTF-59
C
C     COMPUTE DIFFERENCE IN AIR DENSITY                                 TAR00690
C
      HO=DH
```

```
      EQDENS=C.0
      IH=HO-100+EQDENS
       KPT=1.2
      IUP=KPT*IPT
C
C     SAVE WIND COMPONENTS'FOR 1/2 SEC                                  TAR00750
C
      WYR=WY
      WXR=WX
C
C     COMPUTE DIFFERENCE IN PROJECTILE WEIGHT                           TAR00780
C
      IWT=DWT-72
      AMMCTR=0
C
C     SAVE OBSERVED PROJECTILE VELOCITY                                 TAR00810
C
      UC=DSU
C
C     COMPUTE CHANGE IN PROJECTILE VELOCITY                             TAR00830
C
      IU=UO-2500+IUP
      AMMOSH=0
C
C     BALLISTIIC TIME NOTE TBR,TB,TC, VALUES ARE ASSUMPTIONS            TAR00860
C
      TBR=0
      TB=1.0/4.0
      TC=1.0/8.0
C
C     COMPUTE SHIP'S UPDATED HEADING                                    TAR00900
C
      LCQO=CQO/57.293+(TB-TC)*DCQO/57.293
C
C     NOTE CQO AND DCQO CONVERTED TO RADIANS                            TAR00920
C
      SNLCQO=SIN(LCQO)
      CSLCQO=COS(LCQO)
      PDA=0.0
      PDO=0
      PVD=0
C
C     PDO,PDA,PVD ARE THE PARALLAX ARMS                                 TAR00980
C     COMPUTE STATIC GUN PARALLAX                                       TAR00990
C
      PXL=PDA*CSLCQO+PDO*SNLCQO
      PYL=PDO*CSLCQO-PDA*SNLCQO
C
```

```
C     UPDATE OWNSHIP'S POSITION AND REATES TO BALLISTIC TIME              TAR01020
C
      RCXY=ROX+DMOXY*1/4
      DMCYY=DMOY
      RCYY=ROY+DMOYY/4
C
C     ICOUNT=0 INDICATES FIRST PASS FOR TARG3 PRCGRAM                     TAR01060
C
      ICCUNT=0
      QR3=QAR2
      QALR=QALS
      QVR=QAV
C
C     QR3,QRLR,QVR ARE SPOTS FOR AA                                       TAR01110
C
      JCELCK=-1.0/32.0
C
C     JOELCK =-1/32 AN ASSUMPTION                                         TAR01130
C
      ITB=TB-JOELCK
C
C     COMPUTE TARGET POSITION AND RATES UPDATED TC BALLISTIC TIME         TAR01150
C
      RTZL=ZSUBN(1)+CDMRTZ(1)*ITB
      DRTZ=CDMRTZ(1)
      RTXL=ZSUBN(2)+CDMRTZ(2)*ITB
      DRTX=CDMRTZ(2)
      RTYL=ZSUBN(3)+CDMRTZ(3)*ITB
      DRTY=CDMRTZ(3)
      WRITE(6,700) RTZL,RTXL,RTYL
  700 FORMAT(10X,3E15.8)
C
C     NO PREVIOUS BALLISTIC SOLUTION                                      TAR01220
C     COMPUTE RELATIVE TARGET POSITION AT POINT 1                         TAR01230
C
  200 RYL=RTYL-RCYY-PYL
      RXL=RTXL-RCXY-PXL
      WRITE(6,701) ROYY,ROXY,PYL,PXL
  701 FORMAT(10X,4E15.8)
      Y1=RYL**2
      Y2=RXL**2
      RH1=SQRT(Y1+Y2)
      BYL=0
      SIN1=RYL
      COS1=RXL
      CALL ARTAN
      BYL=TAN
      RZ3=RTZL-PVD
```

```
      RHGEN=RH1
      RZGEN=RZ3
      CALL ZLIMIT
      RZL=RZGEN
C
C     CCMPUTE SLANT RANGE TO POINT 1 (RC)                                TAR01390
C
      Y3=RH1**2
      Y4=RZL**2
      WRITE(6,700) RTZL,RTXL,RTYL
      WRITE(6,700) RZL,RXL,RYL
      RC=SQRT(Y3+Y4)
      SNBYL=RZL/RC
      E1=ARSIN(SNBYL)*57.293
      IF(E1-85.GT.0)E1=85
      WRITE(6,803) NEW3DT
 803  FCRMAT(10X,'NEW3DT=',I5)
      WRITE(6,801) BALL1
 801  FCRMAT(10X,'BALL1=',E15.8)
C
C     CHECK FOR NEW TARGET INDICATION(NEWTGT GREATER THAN ZERC)          TAR01460
C
      NEW3DT=0
      IF(NEW3DT)20,20,21
C
C      CHECK FOR BALLISTIC SOLUTION                                      TAR01480
C
  20  IF(BALL1)22,22,21
C     GCCD BALLISTIC SOLUTION SYSTEM IN TRACK MCDE NEXT PROGRAM CALLED   TAR01500
C     IS TARG3                                                           TAR01510
C
C
  22  AQUIRE=0
      BCCUNT=13
      WRITE(6,800) BCOUNT
 800  FORMAT(10X,'BCOUNT=',E15.8)
      WRITE(6,500) RC
 500  FCRMAT(10X,'RC=',E15.8)
      WRITE(6,501)E1
 501  FCRMAT(10X,'E1=',E15.8)
      WRITE(6,502) ITB
 502  FORMAT(10X,'ITB=',E15.8)
      WRITE(6,503) BYL
 503  FORMAT(10X,'BYL=',E15.8)
      WRITE(6,600) RXL,RYL,RZL
 600  FCRMAT(10X,3E15.8)
      RETURN
  21  NEW3DT=1
```

```
C
C     NC BALLISTIC SOLUTICN NEXT PROGRAM CALLEC IS TARG2                TAR01640
C                                                                       TAR00020
      BCCUNT=1
      FSTBAL=1
      WRITE(6,803) NEW3DT
      WRITE(6,800) BCOUNT
      WRITE(6,500) RC
      WRITE(6,501)E1
      WRITE(6,503) BYL
      WRITE(6,502) ITB
      RETURN
    2 BALL1=1
      BCCUNT=0
      RETURN
      END


      SUBROUTINE TARG2
C     THIS SUBROUTINE PERFORMS THEACQUISITION BALLISTICS COMPUTATIONS
C                                                                       TAR00150
C     THIS SUBROUTINE PERFORMS THE ACQUISITION BALLISTICS COMPUTATICNS  TAR00160
C     FOR AA IN CARTESIAN BALLISTICS. ITS FUNCTION IS TO CALCULATE POINTTAR00170
C     3, THE THEORETICAL AIMING PCINT,FROM POINT 1,THE TARGET POSITION  TAR00180
C     AT BALLISTICS TIME. THIS SUBROUTINE DETERMINES THE FIRST ADVANCE  TAR00190
C     PCSITION WHEN NO VALID PRIOR ADVANCE POSITICN EXISTS. THIS        TAR00200
C     SITUATION WOULD OCCUR WHEN EVER,A NEW AA TARGET HAS BEEN ASSIGNED TAR00210
C     THERE WAS NO BALLISTICS SOLUTION DURING THE PREVICUS 1/2 SECCND.  TAR00220
C
      COMMON/RT/COS1,SIN1,TAN
      COMMON/T121/E1,BYL
      COMMON/T13/AARO,RNGDOT,RC
      COMMON/T1/ROX,ROY,DMOX,DMOY,ROXY,RCYY,RXL,RYL,RZL,DMCXY,DMOYY,
     CDRTX,DRTY,DRTZ
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UC
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/T17/BALL1,NEW3DT,NOCLOS,ACUIRE,ICOUNT,KGOOD,BALIND
      COMMON/N2/BCOUNT
      COMMON/CZ/RHGEN,RZGEN
C
      SILFLG=0
C     INCICATES IN ACQUISITION MODE                                     TAR00250
      AQUIRE=1
C     INCICATES MORE THAN CNE ACQUISITION IS REQUIRED BEFORE GUN IS RELETAR00270
C     CCMPUTE TIME OF FLIGHT TO POINT 1                                 TAR00290
      RTEMP=RC
```

```
      TMP3=0
      E3=E1
      ICCDEP=1
      CALL TMCNIT
      T1=TEMP2
C     CARTERIAN BALLISTICS                                              TAR00360
C     COMPUTE TARGET RANGE RATE(DRIBAL)                                 TAR00370
      DRIBAL=(RXL*DRTX+RYL*CRTY+RZL*DRTZ)/RC
C    CHECK IF TARGET IS INCOMING OR OUTBCUND                            TAR00390
      IF(DRIBAL)1,2,2
C     CCMPUTE THE PROJECTILE VELOCITY;U4) TO PCINT4 (PRELIMINARY INTERCETAR00410
C     IF TARGET IS INCOMING THE ACTUAL PROJECTILE VELOCITY IS USED 1/3 CTAR00420
C     FT/SEC TO YD/SEC                                                  TAR00430
  1   U4=U0*1/3
      GC TO 3
C     IF TARGET IS CUTBOUND THE TARGET VELOCITY IS USED AS U4           TAR00460
  2    U4=4/T1*RC
C     COMPUTE TARGET SPEED (DMTBAL)                                     TAR00480
  3   DMTBL1=DRTY**2+DRTX**2+DRTZ**2
      DMTBAL=SQRT(DMTBL1)
C     CCMPUTE TARGET CROSS RANGE RATE (DRL1)                            TAR00510
      DRL2=DMTBAL**2-DRIBAL**2
      DRL1=SQRT(DRL2)
C     CHECK IF PROJECTILE CAN HIT THE TARGET. IF DRL1 IS GREATER THAN U4TAR00540
C     IS TOO FAST THERFORE NO SOLUTION EXIST AND PROGRAM RETURNS TO MAINTAR00550
      IF(DRL1-U4)4,4,5
   5  BALL1=1
      BCCUNT=0
      RETURN
C     DRL1<U4 CMPPUTE LEAD ANGLES                                       TAR00600
  4   SINLS=DRL1/U4
      CCLS=1-SINLS**2
      CCSLS=SQRT(COLS)
C     COMPARE TARGET RANGE RATE WITH CCMPONENT CF PROJECTILE VELOCITY INTAR00640
C     DIRECTICN TO SEE IF TARGET CAN BE HIT                             TAR00650
      U41=U4*CCSLS
      IF(U41-DRIBAL)6,6,7
C     TARGET CANNOT BE HIT CONTROL RETURNED TO MAIN                     TAR00680
  6   BALL1=1
      BCCUNT=0
      RETURN
C     TARGET CAN BE HIT. TIME OF FLIGHT TO POINT 4 IS COMPUTED AND USED TAR00720
C     BEST ESTIMATE FOR POINT 11(T11) THE FIRST ESTIMATE CFINTERCPT TIMETAR00730
C     TIME IS LIMITED TO 64 SECONDS IN AA                               TAR00740
  7   TMP7=U4*COSLS-DRIBAL
      T11=RC/TMP7
      IF(T11-64.GT.0) T11=64
C     CCMPUTE RANGE VALUES AT POINT 11
```

```
      RX11=T11*DRTX+RXL
      RY11=T11*DRTY+RYL
      RZ11=T11*DRTZ+RZL
      RH111=RX11**2+RY11**2
      RH11=SQRT(RH111)
      RHGEN=RH11
      RZGEN=RZ11
      CALL ZLIMIT
      RZ11=RZGEN
      R111=RH11**2+RZ11**2
      R11=SQRT(R111)
C     COMPUTE BEARING ELEVATION, RANGE VALUES TC ADVANCED AIM POINT 3
      SIN1=RY11/RH11
      COS1=RX11/RH11
      CALL ARTAN
      BY3=TAN
      SINE3=RZ11/R11
      IF(SINE3-0.99619.GT.0.0) E3=85/57.293
      E3=ARSIN(SINE3)
      COSE3=COS(E3)
      E3=E3*57.293
      TMP8=IH*0.005+1
      XX=1-IU/U0
      XY=XX*TMP8
      R3=XY*R11
      SINBY3=SIN(BY3/57.293)
      COSBY3=COS(BY3/57.293)
C     COMPUTE HORIZONTAL COMPONENTS OF ADVANCED RANGE
      RX3=R3*COSE3*SINBY3
      RY3=R3*COSE3*COSBY3
      BCCUNT=2
      WRITE(6,804) T1,T11,U4
 804  FORMAT(10X,3E15.8)
      WRITE(6,500) R3
 500  FORMAT(10X,'R3=',E15.8)
      WRITE(6,501) E3
 501  FORMAT(10X,'E3=',E15.8)
      WRITE(6,502) BY3
 502  FORMAT(10X,'BY3=',E15.8)
      RETURN
      END

      SUBROUTINE BCOMPA                                                 BC000010
C                                                                       BC000020
C     THIS SUBROUTINE COMPUTES THE TIME OF FLIGHT TO POINT 3(T3),AND    BC000030
C     THE DERIVATIVES OF T3 WITH RESPECT TO RANGE(DT3DR),ELEVATION(DT3DEBC000040
C      AND THE MIXED PARTIAL(DT3RDE).                                   BC000050
```

```
C                                                                  BC000060
C                                                                  BCC00130
      CCMMON/CA/T3,DT3DR,DT3DE,DT3RDE
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      CCMMON/N2/BCOUNT
      CCMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      CCMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      INTEGER*4 T17                                                GAR00340
C                                                                  TAR00230
C     KCRIND IS THE NUMBEROF COEFFICIENTS PER RANGE INTERVAL
      KORIND=4                                                     BC000030
      ICCDEP=1                                                     BC000040
      TMP3=1                                                       BC000050
      RTEMP=R3                                                     BCC00060
      CALL TMCNIT                                                  BCC00070
      T3=TEMP2                                                     BC000080
      CT3CR=TMP10                                                  BCC00090
      CT3DE=DPDEP                                                  BC000100
      DT3RDE=DPDED                                                 BC000110
      BCCUNT=3                                                     BCC00120
      WRITE(6,100)T3                                               BC000130
 100  FORMAT(10X,'T3=',E15.8)                                      BC000140
      WRITE(6,101)DT3DR                                            BC000150
 101  FCRMAT(10X,'DT3DR=',E15.8)                                   BCC00160
      WRITE(6,102)DT3DE                                            BCC00170
 102  FCRMAT(10X,'DT3DE=',E15.8)                                   BCC00180
      WRITE(6,103)DT3RDE                                           BC000190
 103  FORMAT(10X,'DT3RDE=',E15.8)                                  BCC00200
      RETURN                                                       BC000210
      END                                                          BC000220


      SUBRCUTINE BCOMPB
C                                                                  BCC00020
C     THIS SUBROUTINE FINDS THE SUPERELEVATION ANGLE(EGB).THE VALUE IS BCC00030
C     RETURNED IN TEMP2 IN MINUTES SO MUST BE CHANGED TO DEGREES.THE   BCC00040
C     DERIVATIVE WITH RESPECT TO RANGE (DEGDR) RETURNED IN TMP10,      BCC00050
C     THE DERIVATIVE WITH RESPECT TO ELEVATION (DEGDE) IN DPDEP, THE   BCC00060
C     MIXED PARTIAL (DEGRDE) IN DPDED                                  BCC00070
C                                                                  BCC00080
      CCMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      CCMMON/N2/BCOUNT
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      CCMMON/CB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4,DEGRDE
C                                                                  BC000130
      TMP3=1                                                       BCC00060
      ICCDEP=2                                                     BC000070
```

```
      CALL TMCNIT                                                       BCC00080
      EGB=TEMP2                                                         BCC00090
      EGB1=EGB/60                                                       BCC00100
      E4=E3+EGB1                                                        BCC00110
C     E4=SUM CF E3 AND EGB
      IF(E4-85)1,2,2                                                    BCC00120
  2      E4=85                                                          BCC00130
  1      E41=E4/57.293                                                  BCC00140
      SINE4=SIN(E41)                                                    BCC00150
      CCSE4=CCS(E41)                                                    BCC00160
      DEGDR=TMP10                                                       BCC00170
      DEGDE=DPDEP                                                       BCC00180
      DEGRDE=DPDED                                                      BCC00190
      BCCUNT=4                                                          BCC00200
      WRITE(6,100)EGB1                                                  BCC00210
 100  FORMAT(10X,'EGB1=',E15.8)                                         BCC00220
      WRITE(6,101)E4                                                    BCC00230
 101  FCRMAT(10X,'E4=',E15.8)                                           BCC00240
      WRITE(6,102)DEGDR                                                 BCC00250
 102  FCRMAT(10X,'DEGDR=',E15.8)                                        BCC00260
      WRITE(6,103)DEGRDE                                                BCC00270
 103  FCRMAT(10X,'DEGRDE=',E15.8)                                       BCC00280
      WRITE(6,104)DEGDE                                                 BCC00290
 104  FCRMAT(10X,'DEGDE=',E15.8)                                        BCC00300
      RETURN                                                            BCC00310
      END                                                               BCC00320

      SUBROUTINE BCOMPC                                                 BCCC0010
C                                                                       BCC00020
C     THIS SUBROUTINE COMPUTES THE DERIVATIVE CF RANGE WITH RESPECT TO  BCC00030
C     TC NCN-NCMIMAL INCREMENTS OF PROJECTAL VELCCITY (DRDU) AND        BCC00040
C     THE DERIVATIVES OF DRDU WITH RESPECT TO ELEVATION(DRDUDE). IT ALSOBCC00050
C     COMPUTES THE ACTUAL RANGE EFFECT (URH) DUE TO DRDU                BCC00060
C                                                                       BCC00070
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UC
      CCMMON/N2/BCOUNT
      CCMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      CCMMON/OC/DRDU,DRDUDE,URH
C                                                                       BCC00120
      TMP3=0                                                            BCC00070
      ICODEP=3                                                          BCC00080
      CALL TMCNIT                                                       BCC00090
      DRDU=TEMP2                                                        BCC00100
      URH=DRDU*IU                                                       BCC00110
      DRDUDE=DPDEP
      BCCUNT=5                                                          BCC00120
      WRITE(6,100)DRDU                                                  BCC00130
```

```
  100 FORMAT(10X,'DRDU=',E15.8)                                         BCC00140
      WRITE(6,101)DRDUDE                                                BCC00150
  101 FCRMAT(10X,'DRDUDE=',E15.8)                                       BC000160
      RETURN                                                            BCC00170
      END                                                               BC000180


      SUBROUTINE BCOMPD                                                 BCC00010
C     THIS SUBROUTINE COMPUTES THE DERIVATIVE OF THE TIME OF FLIGHT WITH
C     RESPECT TO THE INCREMENT IN PROJECTILE VELCCITY (DT3DU) AND THE   COM00010
C     TIME OF FLIGHT EFFECT (UT3) DUE TO DT3DU. IT ALSO COMPUTES THE    COM00020
C     MIXED PARTIAL OF TIME OF FLIGHT WITH RESPECT TO INCREMENT IN      COM00030
C     PROJECTILE VELOCITYAND ELEVATION CR RANGE (CT3UDE,DT3UDR)         COM00040
C                                                                       COM00050
C     ELEVATICN OR RANGE (T/DT3UDE,DT3UCR)
      COMMON/T120/IH,IPT,ITF,IU,IWT,HO,UO
      CCMMON/N2/BCOUNT
      CCMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      CCMMON/CD/DT3DU,DT3UDR,CT3UDE,UT3
C
      TMP3=1                                                            BC000070
      ICCDEP=4                                                          BCC00080
      CALL TMCNIT                                                       BC000090
      DT3UDR=TMP10
      CT3DU=TEMP2                                                       BCC00110
      UT3=DT3DU*IU                                                      BC000120
      DT3UDE=DPDEP                                                      BCC00130
      BCCUNT=6                                                          BCC00140
      WRITE(6,100) DT3UDR
  100 FORMAT(10X,'DT3UDR=',E15.8)
      WRITE(6,101)DT3DU                                                 BC000170
  101 FORMAT(10X,'DT3DU=',E15.8)                                        BC000180
      WRITE(6,102)UT3                                                   BC000190
  102 FCRMAT(10X,'UT3=',E15.8)                                          BC000200
      WRITE(6,103)DT3UDE                                                BC000210
  103 FCRMAT(10X,'DT3UDE=',E15.8)                                       BC000220
      RETURN                                                            BC000230
      END                                                               BC000240


      SUBROUTINE BCOMPE                                                 BC000010
C     THIS SUBROUTIME COMPUTES THE DERIVATIVE CF RANGE WITH RESPECT TO  COM00060
C     NCN-NOMINAL INCREMENTS IN AIR DENSITY (DRDH) AND THE DERIVATIVE   COM00070
C     CF DRDH WITH RESPECT TC EVEVATICN (DRDHDE). IT ALSO COMPUTES THE  COM00080
C     ACTUAL RANGE EFFECT DUE TO NON-NOMINAL BALLISTIC DENSITY (HRH).   COM00090
C                                                                       COM00100
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UO
      CCMMON/N2/BCOUNT
```

```
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/CE/DRDH,DRDHDE,HRH
C
      POWDR=0                                                           BC000070
      TMP3=0                                                            BCC00080
      ICODEP=5                                                          BC000090
      CALL TMCNIT                                                       BCC00100
      DRDH=TEMP2                                                        BC000110
      HRH=DRDH*IH                                                       BC000120
      DRDHDE=DPDEP                                                      BC000130
      BCCUNT=7                                                          BC000140
      WRITE(6,100)DRDH                                                  BCC00150
  100 FORMAT(10X,'DRDH=',E15.8)                                         BCC00160
      WRITE(6,101)HRH                                                   BC000170
  101 FORMAT(10X,'HRH=',E15.8)                                          BCC00180
      WRITE(6,102)DRDHDE                                                BC000190
  102 FORMAT(10X,'DRDHDE=',E15.8)                                       BC000200
      RETURN                                                            BCC00210
      END                                                               BC000220


      SUBROUTINE BCOMPF                                                 BCC00010
C     THIS SUBROUTINE COMPUTES THE DERIVATIVE OF TIME OF FLIGHT WITH    COM00110
C     RESPECT TO NON-NOMINAL INCREMENT IN AIR DENSITY (DT3DH),AND       COM00120
C     COMPUTES THE ACTUAL TIME OF FLIGHT EFFECT (HT3) DUE TO DT3DH.     COM00130
C                                                                       COM00140
      COMMON/T120/IH,IPT,ITF,IU,IWT,HO,UO
      COMMON/N2/BCOUNT
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/CF/HT3,DT3DH,DT3HDE,DT3HDR
C
      POWDT=0                                                           BC000070
      TMP3=1                                                            BCC00080
      ICCDEP=6                                                          BC000090
      CALL TMONIT                                                       BC000100
      DT3HDR=TMP10                                                      BCC00110
      DT3DH=TEMP2                                                       BC000120
      HT3=DT3DH*IH                                                      BCC00130
      DT3HDE=DPDEP                                                      BCC00140
      BCCUNT=8                                                          BCC00150
      WRITE(6,100)DT3HDR                                                BCC00160
  100 FORMAT(10X,'DT3HDR=',E15.8)                                       BCC00170
      WRITE(6,101)DT3DH                                                 BC000180
  101 FORMAT(10X,'DT3DH=',E15.8)                                        BCC00190
      WRITE(6,102)HT3                                                   BC000200
  102 FORMAT(10X,'HT3=',E15.8)                                          BC000210
      WRITE(6,103)DT3HDE                                                BC000220
  103 FORMAT(10X,'DT3HDE=',E15.8)                                       BC000230
```

```
      RETURN                                                            BC000240
      END                                                               BC000250


      SUBROUTINE BCOMPG                                                 BC000010
C     THIS SUBROUTINE COMPUTES THE DERIVATIVE OF THE RANGE WITH RESPECT TO A
C     TO NON-NOMINAL INCREMENT IN SURFACE AIR TEMPERATURE (DRDTF) THE   COM00150
C     ACTUAL RANGE EFFECT (TFR3), AND THE MIXED PARTIAL OF RANGE WITH   COM00160
C     RESPECT TO TEMPERATURE AND ELEVATION (DRTFDE).                    COM00170
C                                                                       COM00180
      COMMON/CG/DRTFDE,DRDTF,TFR3
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/N2/BCOUNT
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UC
C
      TMP3=0                                                            BC000070
      ICODEP=7                                                          BC000080
      CALL TMONIT                                                       BC000090
      DRDTF=TEMP2                                                       BC000100
      TFR3=DRDTF*ITF                                                    BC000110
      DRTFDE=DPDEP                                                      BC000120
      BCCUNT=9                                                          BC000130
      WRITE(6,100)DRDTF                                                 BC000140
  100    FORMAT(10X,'DRDTF=',E15.8)                                     BC000150
      WRITE(6,101)TFR3                                                  BC000160
  101 FORMAT(10X,'TFR3=',E15.8)                                         BC000170
      WRITE(6,102)DRTFDE                                                BC000180
  102    FORMAT(10X,'DRTFDE=',E15.8)                                    BC000190
      RETURN                                                            BC000230
      END                                                               BC000240


      SUBROUTINE BCOMPH                                                 BC000010
C     THIS SUBROUTINE COMPUTES THE DERIVATIVE OF T3 WITH RESPECT TO A   COM00190
C     NON-NOMINAL  INCREMENT IN SURFACE TEMPERATURE (DT3DF),COMPUTES THECOM00200
C     ACTUAL TIME OF FLIGHT EFFECT (TFT3) DUE TO DT3DTF,CALL THE WEIGHT COM00210
C     EFFECT SUBROUTINE (RTWT), PERFORMS A PARTIAL SUM OF ALL  THE      COM00220
C     RANGE EFFECTS AND TIME OF FLIGHT EFFECTS CALCULATED SO FAR,       COM00230
C     AND CALCULATES THE DERIVATIVE OF TIME OF FLIGHT TO POINT 2        COM00240
C     (THE PREDICTED IMPACT POINT)WITH RESPECT  TO RANGE (DT2DR)        COM00250
C                                                                       COM00260
      COMMON/CA/T3,DT3DR,DT3DE,DT3RDE
      COMMON/CB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4,DEGRDE
      COMMON/CC/DRDU,DRDUDE,URH
      COMMON/CD/DT3DU,DT3UDR,DT3UDE,UT3
      COMMON/CE/DRDH,DRDHDE,HRH
      COMMON/CF/HT3,DT3DH,DT3HDE,DT3HDR
      COMMON/CG/DRTFDE,DRDTF,TFR3
```

```
      CCMMON/T21/RTEMP,R3,RX3,RY3,RHOLC
      COMMON/T120/IH,IPT,ITF,IU,IWT,HO,UO
      CCMMON/N2/BCOUNT
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      CCMMON/CH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
C
      PCWDT=0
      PCWDR=0
      TMP3=0                                                            BC000050
      ICODEP=8                                                          BCC00060
      CALL TMCNIT                                                       BCC00070
      DT3DTF=TEMP2                                                      BC000080
      TFT3=DT3DTF*ITF                                                   BC000090
      DT3FDE=DPDEP                                                      BC000100
      CALL RTWT                                                         BCC00110
C     CCMPUTE FIRST PARTIAL SUM OF RANGE EFFECTS WITHOUT R3
      J1R3=WTR3+URH+HRH+TFR3+POWDR                                      BC000120
C     COMPUTE FIRST PARTIAL SUM OF TIME OF FLIGHT WITH T3               CCM00270
      R2J1=J1R3+R3                                                      BCC00130
C     CCMPUTE FORST PARTIAL SUM OF TIME CF FLIGHT WITH T3
      T2J1=WTT3+UT3+HT3+TFT3+POWDT+T3
C     CCMPUTE DERIVATIVE CF TIME CF FLIGHT TO PCINT 2 WITH              COM00280
C     RESPECT TO RANGE                                                  COM00290
      DT2DR=DT3DR+DT3HDR*IH+DT3UDR*IU                                   BC000150
      BCCUNT=10                                                         BCC0016C
      WRITE(6,101)TFT3                                                  BC000190
  101 FCRMAT(10X,'TFT3=',E15.8)                                         BCC00200
      WRITE(6,102)DT3FDE                                                BC000210
  102 FORMAT(10X,'DT3FDE=',E15.8)                                       BCC00220
      WRITE(6,100)DT3DTF                                                BCOC0170
  100 FCRMAT(10X,'DT3DTF=',E15.8)                                       BC000180
      RETURN                                                            BCC00200
      END                                                               BC000210


      SUBROUTINE BCOMPI                                                 BC000010
C     THIS SUBROUTINE CCMPUTES THE DOWNRANGE ANC CRCSSRANGE COMPONENTS  COM00310
C     OF GUN MOTION (DMR30G AND DMBOG) BEFORE CALLING GUNMCT. IT THEN   COM00320
C     CCMPUTES THE TANGENT CF ANGLE OF FALL (TANW) BEFORE CALLING       CCM00330
C     THE WIND ECCECTS(WIND) SUBROUTINE                                 COM00340
C                                                                       CCM00350
      CCMMON/T1/ROX,ROY,DMCX,DMOY,ROXY,ROYY,RXL,RYL,RZL,DMCXY,DMCYY,
     CDRTX,DRTY,DRTZ
      CCMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      CCMMON/CB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4,DEGRDE
      COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
```

```
      COMMON/ I/ANGLEW,DMR30G,DMBOG,SINEGB,COSEGB,TANEG,
     CTANW,TOVERW
      CCMMON/CH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,CT3DTF
      CCMMON/N2/BCOUNT
C                                                                       COM00300
C     CCMPUTE DOWNRANGE COMPONENT OF GUN MOTION
      DMR30G=DMOXY*SINBY3+DMOYY*COSBY3                                  BC000040
C     CCMPUTE CROSSRANGE COMPONENT OF GUMMOTION
      DMBOG=DMOXY*COSBY3-DMOYY*SINBY3                                   BC000050
      SINEGB=SIN(EGB1/57.293)
      COSEGB=COS(EGB1/57.293)
      CALL GUNMOT                                                       BCC00070
      TMP3=0                                                            BCC00080
C     RANGE IS SET EQUAL TO FIRST PARTIAL SUM OF RANGE EFFECTS BEFORE   COM00360
C     TMCNIT IS CALLED                                                  COM00370
      RTEMP=R2J1                                                        BC000090
      ICCDEP=9                                                          BCC00100
      CALL TMCNIT                                                       BC000110
      ANGLEW=TEMP2                                                      BCC00120
C      CCMPUTE TANGENT OF GUN ELEVATICN (TANEG) MUST CHECK FOR INFINITY
      IF(COSEGB.EQ.0) TANEG=500000000                                   BC000140
      TANEG=SINEGB/COSEGB                                               BC000150
C     CCMPUTE TANGENT OF ANGLE OF FALL(TANW)
      TANW=TANEG*ANGLEW                                                 BC000160
      IF(TANW.GT.10000) TOVERW=0                                        BCC00170
      TCVERW=T2J1/TANW                                                  BC000180
      IF(TOVERW.GT.100000) TCVERW=10000
      CALL WIND                                                         BC000190
      BCCUNT=11                                                         BCC00200
      WRITE(6,100)ANGLEW                                                BCC00210
  100 FCRMAT(10X,'ANGLEW=',E15.8)                                       BC000220
      WRITE(6,101)TCVERW                                                BC000230
  101 FCRMAT(10X,'TOVERW=',E15.8)                                       BCC00240
      RETURN                                                            BC000250
      END                                                               BCC00260

      SUBROUTINE BCOMPJ
C     THIS SUBROUTINE OBTAINS A FIRST PARTIAL SUM OF RANGE EFFECTS      COM00390
C     AND A SECOND PARTIAL SUM OF TIME OF FLIGHT EFFECTS AND CALLS      COM00400
C     CORLIS TO OBTAIN COROILIS EFFECTS                                 COM00410
C                                                                       COM00420
      CCMMON/N2/BCCUNT
      CCMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      CCMMON/CW/DRDWRH,DTCWRH,WBG,WRHG,WRLS,WR3,WT3
      CCMMON/CGU/DRDMEY,DRDMRY,DT3DMY,CT3MEY,MR3CR3,MR30T3,MOLS,MORLS
      CCMMON/CH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
```

```
     C,DT3DTF
      COMMON/CJ/J2R3,R2J2,T2J2
C                                                                       COM00380
C     COMPUTE RANGE EFFECTS DUE TO WIND AND GUNMOTION
      J2R3=WR3+MR3OR3
C     COMPUTE FIRST PARTIAL SUM OF RANGE EFFECTS INCLUDING R3
      R2J2=R2J1+J2R3
C     MODIFY RTEMP FOR TMONIT
      RTEMP=R2J2
C     COMPUTE SECOND PARTIAL SUM OF TIME OF FLIGHT BY ADDING EFFECTS    COM00430
C     OF WIND AND GUN MOTION TO FIRST PARTIAL SUM                       COM00440
      T2J2=T2J1+WT3+MR3OT3
      CALL CORLIS
      BCOUNT=12
       WRITE(6,100) RTEMP
  100 FORMAT(10X,'RTEMP=',E15.8)
      WRITE(6,101) T2J2
  101 FORMAT(10X,'T2J2=',E15.8)
      RETURN
      END


      SUBROUTINE BCOMPK                                                 BCC00050
C                                                                       COM00450
C     THIS SUBROUTINE COMPUTES THE INTERCEPT GEOMETRY BY DETERMINING    COM00460
C     THE HORIZONTAL RANGE, VERICAL RANGE, SLANT RANGE AND BEARING      COM00470
C     TO THE PREDICTED PROJECTILE IMPACT POINT. IT ALSO COMPUTES        COM00480
C     THE ELEVATION ANGLES FOR THE GUN                                  COM00490
      COMMON/RT/COS1,SIN1,TAN
      COMMON/OW/DRDWRH,DTDWRH,WSG,WRHG,WRLS,WR3,WT3
      COMMON/CGU/DRDMEY,DRDMRY,DT3DMY,DT3MEY,MR3CR3,MR3CT3,MOLS,MORLS
      COMMON/OCR/KRLS,KR3,KTRRE3,KT3,T2J2SQ,TANE4
      COMMON/OJ/J2R3,R2J2,T2J2
      COMMON/CH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
      COMMON/T24/QALS,QAR2,QAV,ERVG,QR3,QRLR,QVR,RV
      COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/CDR/DORLS
      COMMON/N2/BCOUNT
      COMMON/CK/R2,E2,T2,BY2
C                                                                       COM00500
      REAL * 4 KRLS,KR3,KTRRE3,KT3
      CALL DRIFT                                                        BCO00060
C     COMPUTE TOTAL CROSS RANGE EFFECTS
      BRLS=DORLS+KRLS+WRLS+MORLS
C     COMPUTE TIME OF FLIGHT TO IMPACT POINT 2
      T2=T2J2+KT3                                                       BCC00070
```

```
C       COMPUTE VERTICAL RANGE EFFECTS TO POINT 2
        BR3=J1R3+J2R3+KR3                                                BC000080
        BR3Q=BR3-QR3                                                     BC000090
        RV2=(R3+BR3Q)*SINE3                                              BC000100
        WRITE(6,500) RV2,BR3Q
C       COMPUTE HORIZONTAL RANGE TO IMPACT POINT 2                       COM00510
        BRH3=BR3Q*COSE3                                                  BC000110
        BRX3=BRH3*SINBY3+BRLS*COSBY3                                     BC000120
        BRY3=BRH3*COSBY3-BRLS*SINBY3                                     BC000130
        RY2=RY3+BRY3                                                     BC000140
        RX2=RX3+BRY3                                                     BC000150
        RH2=SQRT(RX2**2+RY2**2)                                          BC000160
C       COMPUTE SLANT RANGE TO IMPACT POINT 2
        R2=SQRT(RV2**2+RH2**2)                                           BC000170
C       COMPUTE BEARING TO IMPACT POINT 2
        SIN1=RY2
        COS1=RX2
        CALL ARTAN
        BY2=TAN
C       COMPUTE ELEVATION TO IMPACT POINT 2
        SINE2=RV2/R2                                                     BC000190
        IF(SINE2-0.9962.GT.0.0) E2=85                                    BC000200
        E21=ARSIN(SINE2)                                                 BC000210
        E2=E21*57.293
        BCOUNT=13                                                        BC000230
        WRITE(6,100)T2                                                   BC000240
 100    FORMAT(10X,'T2=',E15.8)
        WRITE(6,101) R2                                                  BC000260
 101    FORMAT(10X,'R2=',E15.8)
        WRITE(6,102) E2                                                  BC000280
 102    FORMAT(10X,'E2=',E15.8)
        WRITE(6,103) BY2
 103    FORMAT(10X,'BT2=',E15.8)
        WRITE(6,105) BRLS
 105    FORMAT(10X,'BRLS=',E15.8)
        WRITE(6,500) T2J2,KT3
 500    FORMAT(10X,2E15.8)
        RETURN                                                           BC000300
        END                                                              BC000310

        SUBROUTINE TARG3
C                                                                        COM00520
C       THIS SUBROUTINE PERFORMS THE ITERATION AND CLOSURE COMPUTATIONS  COM00530
C       FOR AA CATESIAN BALLISTICS. ITS FUNCTION IS TO UPDATE THROUGH    COM00540
C       TWO ITERATIONS THE PREVIOUS ADVANCED AIM POINT (3) SUCH  THAT    COM00550
C       SATISFACTORY COINCIDENCE IS OBTAINED BETWEEN THE PREDICTED IMPACT COM00560
C       POINT (2) AND THE PREDICTED TARGET POSITION AT IMPACT TIME (21)  COM00570
```

```
C     THE CLOSURE EQUATIONS DETERMINE WHEN THERE IS SUFFICIENT CLOSENESSCOM00580
C     BETWEEN THE TWO POINTS FOR AN ACCEPTABLE SOLUTION. THE INPUT      COM00590
C     VALUES TO TARG3 COME FROM TWO SOURCES. IN THE TRACKING MODE       COM00600
C     THE DATA FROM THE PREVIOUS PASS THROUGH TARG3 TO BCOMPK IS        COM00610
C     USED TO UPDATE CURRENT VALUES. IN ACQUISITION MODE THE VALUES     COM00620
C     JUST CALCULATED IN TARG2 TO BCOMPK ARE USED                       COM00630
C                                                                       COM00640
      COMMON/RT/COS1,SIN1,TAN
      COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/N2/BCOUNT
      COMMON/T31/BY22,IBY22,E22,R22,IT2
      COMMON/T3/RX21,RY21,DMRT21,DE21,DBY21,E21
      COMMON/OH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
      COMMON/CA/T3,DT3DR,DT3DE,DT3RDE
      COMMON/OD/DT3DU,DT3UDR,DT3UDE,UT3
      COMMON/CF/HT3,DT3CH,DT3FDE,DT3HDR
      COMMON/T121/E1,BYL
      COMMON/T1/ROX,ROY,DMOX,DMOY,ROXY,RCYY,RXL,RYL,RZL,DMCXY,DMOYY,
     CDRTX,DRTY,DRTZ
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UC
      COMMON/T18/AARO,RNGDOT,RC
      COMMON/T17/BALL1,NEW3DT,NOCLOS,AQUIRE,ICOUNT,KGOOD,BALIND
      COMMON/CK/R2,E2,T2,BY2
      COMMON/OZ/RHGEN,RZGEN
C
      REAL*4 IR21
      REAL*4 IT2,IT2X
C
      SILFLG=0
C     CARTESIAN BALLISTIC
C     COMPUTE TARGET EASTING AND NORTHING COMPONENTS AT IMPACT TIME BY  COM00650
C     UPDATING COMPONENTS AT POINT 1                                    COM00660
      RX21=DRTX*T2+RXL
      RY21=DRTY*T2+RYL
      X=RX21**2
      Y=RY21**2
C     COMPUTE HORIZONTAL RANGE TO TARGET AT IMPACT TIME
      RH21=SQRT(X+Y)
C      COMPUTE VERTICAL RANGE TO TARGET AT IMPACT TIME BY UPDATING POINT 1
      RHGEN=RH21
      RZ21=RZL+T2*DRTZ
      RZGEN=RZ21
      CALL ZLIMIT
      RZ21=RZGEN
      X1=RZ21**2
      Y1=RH21**2
```

```
      R21=SQRT(X1+Y1)
      IF(SILFLG)1,1,2
C     COMPUTE SINE AND COSINE OF BEARING TO TARGET POSITICN
    1 SNBY21=RX21/RH21
      CSBY21=RY21/RH21
C     COMPUTE TARGET DOWNRANGE RATE DMRT21 AND TARGET CROSSRANGE       COM00670
C     RATE DMBT21                                                      COM00680
      TMP=DRTX*SNBY21
      DMRT21=DRTY*CSBY21+TMP
      TMP1=DRTY*SNBY21
      DMBT21=DRTX*CSBY21-TMP1
C     COMPUTE BEARING AND ELEVATICN TO POINT 21 LIMIT ELEVATION TO 85
    2 SIN1=RY21
      CCS1=RX21
      CALL ARTAN
      BY21=TAN
      X=RZ21/R21
      X1=X-0.9862
      IF(X1)3,4,4
    4 E21=85
      GC TO 5
    3 E21=ARSIN(X)*57.293
C     CCMPUTE DIFFERENCE BETWEEN RANGE TO IMPACT POINT 21 ANDTARGET    COM00690
C     RANGE AT IMPACT TIME                                             COM00700
    5 IR21=R21-R2
      TMP=RX21*DRTX
      TMP1=RY21*DRTY+TMP
      TEMP2=DRTZ*RZ21
C     CCMPUTE TARGET RANGE RATE DMRT21
      DMRT21=(TEMP2+TMP1)/R21
C     CCMPUTE TARGET HORIZONTAL RATE AL
      AL=TMP1/RH21
C     CCMPUTE TARGET VERTICAL RATE DRET21
      AL1=AL*RZ21
      AL2=RH21*DRTZ-AL1
      DRET21=AL2/R21
C     COMPUTE DERIVATIVE OF IMPACT TIME WITH RESPECT  TO ELEVATION DT2DE
      DT2DE=DT3DE+IU*DT3UDE+IH*DT3HDE
C     CCMPUTE MIXED PARTIAL WITH RESPECT TO RANGE AND ELEVATION DT2CRE
      DT2RDE=(DT2DE*57.293)/R2
      TMP=DT2RDE*DRET21
      TMP1=DMRT21*DT2DR+TMP
      DENOM=1-TMP1
      TMP2=IR21*DT2DR
      TMP3=E21-E2
      TMP4=TMP3*0.0174533
      XNUMER=TMP4*DT2RDE*R2+TMP2
C     CCMPUTE IT2 TIME INCREMENT TO T2
```

```
      IT2=XNUMER/DENOM
      WRITE(6,700) DT2DR,DT2DE
      WRITE(6,700) DT2RDE,TMP3
      WRITE(6,700) DENOM,XNUMER
  700 FORMAT(10X,2E15.8)
      IT2X=ABS(IT2)
      IT2X1=IT2X-64
      IF(IT2X1)11,12,12
   12 IF(IT2)13,11,15
   13 IT2=-64
      GO TO 11
   15 IT2=64
C     MUST LIMIT IT2 BETWEEN -64 AND +64 SEC. FIRST
C     THE ABSOULTE VALUE OF TI2 IS SUB FROM 64 IF THE
C     RESULT IS POSITIVE A CHECK IS MADE ON SIGN OF IT2
C     IF IT2 IS + IT2 IS MADE EQUAL TC +64 ,IF - TC -64
   11 IF(AQUIRE)21,21,22
C     TRACK MCDE
C     NOW LIMIT IT2 TO 4 OR - 8 SEC
   21 IT2X2= IT2X-8
      IF(IT2X2)22,24,24
   24 IF(IT2)25,22,27
   25 IT2=-8
      GC TO 22
   27 IT2=8
C     ACQUISITION MODE OR CONTINUE TRACK MODE
C     COMPUTE ESTIMATED INTERCEPT SLANT RANGE R22
   22 R22=R21+IT2*DMRT21
C     CARTESIAN BALLISTICS
      TMP=IT2*DRET21/R22
      WRITE(6,700)IT2,IR21
      TMP1=TMP*57.293
      TMP11=ABS(TMP1)
      TMP12=TMP11-45
      IF(TMP12)41,42,42
   42 IF(TMP1)43,41,44
   43 TMP1=-45
      GC TO 41
   44 TMP1=45
C     CCMPUTE ELEVATION TO PREDICTED INTERCEPT PCINT
   41 E22=E21+TMP1
      EE22=E22-85
      IF(EE22)51,52,52
   52 E22=85
   51 EEE22=E22+5
      IF(EEE22)53,53,54
   53 E22=-5
C     NOW E22 IS LIMITED BETWEEN -5 AND +85 DEGREES
```

```
   54 SINE22=SIN(E22/57.293)
      RH22=R22*SINE22
      TMP=IT2*DMBT21
      TMP1=TMP/RH22
      TMP2=TMP1*57.293
C     CCMPUTE BEARING TO PREDICTED INTERCEPT PCINT
      BY22=BY21+TMP2
      BY221=BY22-360
      IF(BY221)61,62,63
   62 BY22=0
      GC TO 61
   63 BY22=BY22-360.0
C     CHECK IF THIS IS THE FIRST OR SECCND PASS THRU TARG3
   61 IF(ICOUNT)101,101,102
C     FIRST PASS
C     ADVANCE RANGE IS UPDATED BASED CN IMPACT PCINT 2 ANDESTIMATED      COM00710
C     INTERCEPT POINT 22                                                 COM00720
  101 R3=(R3*R22)/R2
C     CCMPUTE NEW ADVANCE BEARING BY3 AND ITS SINE AND COSINE
      BY5=BY3+BY22-BY2
      BY5=BY5-360
      IF(BY5)103,104,105
  104 BY3= 0.0
      GC TO 106
  105 BY3=360-BY5
      GC TO 106
  103 BY3=BY5+360
  106 SINBY3=SIN(BY3/57.293)
      CCSBY3=COS(BY3/57.293)
C     CCMPUTE NEW ADVANCE ELEVATION E3 AND ITS SINE AND COSINE
      E7=E3-E2
      IF(E7-30.GT.0) E7=30
      IF(E7+30.LT.0) E7=-30
      E3=E22+E7
C     LIMIT E3 BETWEEN -5 AND 85
      IF(E3-85.GT.0) E3=85
      IF(E3+5.LT.0) E3=-5
      SINE3=SIN(E3/57.293)
      COSE3=COS(E3/57.293)
      TEMP=R3*COSE3
C     CCMPUTE HORIZONTAL COMPONENT OFADVANCE PCSITION RX3 AND RY3
      RX3=TEMP*SINBY3
      RY3=TEMP*CCSBY3
C     THIS PASS COMPLETED NEXT PROGRAM CALLED IS BCCMPA
      ICCUNT=1
      BCCUNT=2
      WRITE(6,100) RX21
  100 FCRMAT(10X,'RX21=',E15.8)
```

```
      WRITE(6,301) RY21
 301  FORMAT(10X,'RY21=',E15.8)
      WRITE(6,302) RZ21
 302  FORMAT(10X,'RZ21=',E15.8)
      WRITE(6,303) R21
 303  FORMAT(10X,'R21=',E15.8)
      WRITE(6,304) BY21
 304  FORMAT(10X,'BY21=',E15.8)
      WRITE(6,305) E21
 305  FORMAT(10X,'E21=',E15.8)
      WRITE(6,306) DMBT21
 306  FORMAT(10X,'DMBT21=',E15.8)
      WRITE(6,307) DMRT21
 307  FORMAT(10X,'DMRT21=',E15.8)
      WRITE(6,108) E22
 108  FORMAT(10X,'E22=',E15.8)
      WRITE(6,109) BY22
 109  FORMAT(10X,'BY22=',E15.8)
      WRITE(6,110) R22
 110  FORMAT(10X,'R22=',E15.8)
      WRITE(6,111) DRET21
 111  FORMAT(10X,'DRET21=',E15.8)
C     CHECK FOR CLOSURE
 500  AL=IR21*NOCLOS-50
      NOCLOS=0
C     CHECK IF RANGE LIMIT VIOLATED,VIOLATED IF AL GREATER THAN 0
      IF(AL.GT.0) NOCLOS=AL
C     CHECK IF IT2 IS GREATER THAN IT2TST IF SO NO CLOSURE              COM00730
      AL1=ABS(IT2)
      IF(AQUIRE.GT.0) IT2TST=32
      IT2TST=2
      AL2= AL1-IT2TST
      IF(AL2.GT.0) NOCLOS=AL2
C     CHECK IF INCOMING OR OUTBOUND TARGET, IF INCOMING CLOSURE IS GOOD
      IF(DMRT21)201,201,202
  201 RETURN
  202 AL3=DMRT21*DT2DR/2
C     OUTBOUND TARGET AND STILL CAN HIT IT GOOD CLOSURE
      IF(AL3-1.LT.0) RETURN
      IF(IR21.LT.0) RETURN
C     IF TARGET IS OUTBOUND AND IR21 IS + CANNOT HIT TARGET NO CLOSURE
      NOCLOS=IR21
      WRITE(6,800) NOCLOS
 800  FORMAT(10X,'NOCLOS=',E15.8)
      RETURN
C     SECOND PASS
C     COMPUTE DIFFERENCE IN R22 AND R2
  102 IR22=R22-R2
```

```
C       CCMPUTE DIFFERE CE IN BY22 AND BY2
        IBY22=BY22-BY2
C       NEXT PROGRAN TO CALL IS TARG4
        WRITE(6,100) RX21
        WRITE(6,301) RY21
        WRITE(6,302) RZ21
        WRITE(6,303) R21
        WRITE(6,304) BY21
        WRITE(6,305) E21
        WRITE(6,306) DMBT21
        WRITE(6,307) DMRT21
        WRITE(6,108) E22
        WRITE(6,109) BY22
        WRITE(6,110) R22
        WRITE(6,111) DRET21
        BCCUNT=14
        GC TO 500
        END


        SUBROUTINE TARG4
C       THIS SUBROUTINE COMPUTES THE BALLISTICS OLTPUTS FOR AA
C       CARTESIAN BALLISTICS. IT COMPUTES THE FINAL INERTIAL GUN
C       CRDERS, GUN ORDER RATES, AND FUZE TIME RATE, AS WELL AS
C       THE INERTIAL IMPACT COORDINATES.  THESE COMPUTATIONS ARE
C       ORIENTED TO THE NEXT 1/16 SECONC. TARG4 ALSC PROVIDES           OPS00010
C       TIME OF FLIGHT FOR DISPLAY ON GCC, CLOSURE CHECKS,AND
C       CCNTROL OF THE BALLISTICS SOLUTICN LIGHTS ON THE GCC.
C
        CCMMON/OH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
       C,CT3DTF
        CCMMON/CB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4,CEGRDE
        CCMMON/CK/R2,E2,T2,BY2
        CCMMON/T19/TB,TBR
        CCMMON/ODR/DORLS
        CCMMON/T24/QALS,QAR2,QAV,ERVG,QR3,QRLR,QVR,RV
        CCMMON/T1/ROX,ROY,DMCX,DMCY,ROXY,RCYY,RXL,RYL,RZL,CMOXY,DMOYY,
       CDRTX,DRTY,DRTZ
        CCMMON/T21/RTEMP,R3,RX3,RY3,RHOLC
        CCMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
        CCMMON/T31/BY22,IBY22,E22,R22,IT2
        CCMMON/T3/RX21,RY21,DMRT21,DE21,DBY21,E21
        CCMMON/T17/BALL1,NEW3DT,NOCLOS,ACUIRE,ICCUNT,KGOOD,BALIND
        CCMMON/T4/BGYAUP,BTT2,DBGY,DEG,DT216,DT5,EGAUP,FLASH,LITSET,GE,QLS,TT2
        CCMMON/N2/BCOUNT
        REAL *4 IT2
C                                                                       OPS00020
        KGCOD=0
```

```
C     FIRST A CHECK IS MADE TO DETERMINE IF CLCSURE HASOCCURED
      IF (NOCLOS)1,1,2
C     CLCSURE HAS OCCURRED
  1   IF(KGOOD)3,3,4
  3   AL=0
      GC TO 10
  4   KGCOD=KGCOD-1
      DT5=0
      AL=1
      GC TO 10
C     CLOSURE HAS NOT OCCURRED
  2   IF(AQUIRE)5,5,6
C     ACQUISITICN MODE
  6   BALL1 =1
      BCCUNT=0
      RETURN
C     NO BALLISTIC SOLUTION (BALL1=1)
C     TRACK MCDE
  5   B2=1
      AL=DMRT21+344.61
      IF(AL.LT.0.0) GO TO 8
      B2=B2-1
      AL=DMRT21+84.41
      IF(AL.LT.0.0) GO TO 8
      B2 =-1
  8   B2= B2+1
      AL=664
  7   AL=AL/8
      IF(B2.EQ.0.0) GO TO 4
      B2=B2-1
      GC TO 7
C     START OF TARG4 COMPUTATIONS
 10   IF(AL.GT.0.0) GO TO 50
C     CCMPUTE SPCT VALUES IN MILS
      CLS= QRLR
      QE=QVR
C     CCMPUTE FINAL GUN ELEVATION(E4BT)
      TMP=(R22-R2)
      TEMP=TMP*DEGDR
      TEMP1= E22-E2
      TEMP2=TEMP1*DEGDE
      TEMP4=TEMP2+TEMP1+TEMP+E4
      E4B =TEMP4+QE*0.05625
      WRITE(6,700) TMP,TEMP
      WRITE(6,700) TEMP1,TEMP2
      WRITE(6,700) TEMP4,E4B
      IF(E4B-85)11,12,12
 12   E4BT=85
```

```
      GC TO 20
   11 E4BT=E4B
C     COMPUTE FINAL GUN BEARING(BTBG4)
 20   QLS1=QLS*0.05625
      BTBGY=QLS1+BY3+IBY22
      IF(BTBGY-360)13,14,15
   13 BTBGY=BTBGY
      GC TO 30
   14 BTBGY=0.0
      GC TO 30
   15 BTBGY=BTBGY-360
C     COMPUTE THE PREDICTED TIME OF FLIGHT AT BALLISTIC TIME(BTT2)
   3C BTT2=IT2+T2
      WRITE(6,700) IT2,T8
 70C  FCRMAT(10X,2E15.8)
      AL=BTT2-240
      IF(AL.GT.0.0)DISPT2=240
      DISPT2=AL
C     CARTFSIAN BALLISTICS
      DRY=DRTY-DMOYY
      DRX=DRTX-DMOXY
      BY221=BY22/57.293
      SNBY22=SIN(BY221)
      CSBY22=COS(BY221)
      TMP=DRX*SNBY22
      TMP1=DRY*CSBY22+TMP
      E221=E22/57.293
      TMP2=SIN(E221)
      TMP3=COS(E221)
      TMP4=TMP3*TMP1
C     CCMPUTE SLANT RANGE RATE(DR22)
      DR22=DRTZ*TMP2+TMP4
      TMP6=TMP1*TMP2
      TMP7=TMP1*TMP2
      TMP7=DRTZ*TMP3-TMP6
C     TMP7=APPARENT ELEVATION RATE
C     CCMPUTE APPARENT CRCSS RANGE RATE (DRLS22)
      TMP8=DRY*SNBY22
      TMP9=DRX*CSBY22
      DRLS22=TMP9-TMP8
C     COMPUTE GUN BEARING RATE (DBGY)
      TMP10=TMP3*R22
      TMP11=TMP10*2*3.14159
      DBGGY=DRLS22/TMP11
C     CCMPUTE FUSE TIME RATE (DT5)
      TEMP11=(1/DT2DR)-DR22
      DT5= DR22/TEMP11
      DT216=DT5/16
```

```
C       CCMPUTE UPDATED RANGE RATE (DR2)
        DR2=DR22*(1+DT5)
C       CCMPUTE UPDATED GUN ELEVATION RATE(DEG)
        TMP12=DEGDR*DR2
        TMP13=TMP7*(1+DT5)
        TEMP14=R22*2*3.14159
        TMP14=TMP13/TEMP14
        DEG=TMP12+TMP14
C       CCMPUTE UPDATED GUN BEARING RATE (DBGY)
        CBGY=DBGGY*(1+DT5)
        AL=E4BT-85
        IF(AL)31,32,32
   32   DEG=0
  31    TBFIN=15.0/32.0
C       VALUE OF TBFIN IS ASSUMED
        DELTMP=TBFIN-TB
        WRITE(6,700) TB,TBFIN
C       CCMPUTE ADJUSTED TIME OF FLIGHT
        TT2=BTT2+DELTMP*DT5
C       AA MODE, CALCULATE FINAL GUN ELEVATION ORDERS(EGAUP) AND
C       FINAL GUN BEARING CRDERS(BGYAUP)
        AL=DELTMP*DEG
        AL=AL+E4BT
        AL1=AL-85
        IF(AL1)41,42,42
   42   EGAUP=85
        DEG=0
        GO TO 60
  41    EGAUP=AL
  60    BGYAUP=DELTMP*DBGY+BTBGY
        BGYAP=BGYAUP-360
        IF(BGYAP)71,72,73
   73   BGYAUP=BGYAUP-360.0
        GO TO 71
   72   BGYAUP=0.0
   71   IF(T2-24.6)75,76,76
   76   FLASH=200
        GC TO 77
   75   LITSET=200
   77   BALIND=1
   50   BALL1=0
        BCCUNT=C
        WRITE(6,100) DELTMP
  100   FCRMAT(10X,'DELTMP=',E15.8)
        WRITE(6,101) BTT2
  101   FCRMAT(10X,'BT22=',E15.8)
        WRITE(6,102) E4BT
  102    FORMAT(10X,'E4BT=',E15.8)
```

```
      WRITE(6,103) DBGY
 103  FORMAT(10X,'DBGY=',E15.8)
      WRITE(6,104) DEG
 104  FORMAT(10X,'DEG=',E15.8)
      WRITE(6,105) EGAUP
 105  FORMAT(10X,'EGAUP=',E15.8)
      WRITE(6,106) BGYAUP
 106  FORMAT(10X,'BGYAUP=',E15.8)
      RETURN
      END


      SUBROUTINE CORLIS
C     THIS SUBROUTINE COMPUTES BALLISTIC CORIOLIS EFFECTS FOR SLANT      OPS00030
C     RANGE ALTITUDE, TIME OF FLIGHT, CROSSRANGE AND DOWNRANGE EFFECTS.  OPS00040
C     KORIND IS SET TO 6 TO INDICATE TO TMONIT THAT CORLIS IS A NON-     OPS00050
C     STANDARD TABLE WITH 6 COEFFICIENTS FOR 3 LINEAR FUNCTIONS. THIS    OPS00060
C     ALSO INDICATES THAT TMONIT SHOULD USE THE SEARCH SUBROUTINE WHICH  OPS00070
C     FINDS THE CORRECT RANGE PARTITION ,COEFFICIENT  ADDRESS IN         OPS00080
C     BALLISTICS TABLES AND NOT THE POLYN SUBROUTINE . TWO SETS OF       OPS00090
C     COEFFICIENTS ARE EVALUATED FOR THE AA MODE                         OPS00100
C                                                                        OPS00110
      COMMON/CB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4
      COMMON/I/ANGLEW,DMR30G,DMB0G,SINEGB,COSEGE,TANEG,
     CTANW,TOVERW
      COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
      COMMON/CJ/J2R3,R2J2,T2J2
      COMMON/OA/T3,DT3DR,DT3DE,DT3RDE
      COMMON/T12/SINLAT,COSLAT,CQ0,DCQC,SINLAR,COSLAR
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/N1/ICODEP,DPCEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      COMMON/CCR/KRLS,KR3,KTRRE3,KT3,T2J2SQ,TANE4
      DIMENSION KORTP(6)
      INTEGER*4 B5
      REAL*4 IR3
      REAL *4 KORTP
      REAL * 4 KGYP,KBYP,KEYP,KYP
      REAL * 4 KRLS,KR3,KTRRE3,KT3
C
      XNEWOK=7.2921 E-05
      KORIND=6
      ICODEP=10
      TMP3=0
      CALLTMONIT
      B6=0
C     B8=0 ALLOWS FOR TWO PASSES THRU STACKS
```

```
      B8=0
C     L=1 INDICATES FIND THE HIGHEST STACK COEFFIENTS
      L=1
C     K IS THE HIGHEST ELEVATION STACK
      K1=K
C     B5 IS THE ADDRESS OF FIRST COEFFICIENT
 10   B5=HOLDB3(L)
C     COMPUTE KGYP,DEYP,DBYP FOR EACH STACK THESE VALUES ARE STORED IN MATRIX KO
C     KORPT
      KORTP(B6+3)=TABLE(ICODEP,K1,B5+1)*IR3+TABLE(ICODEP,K1,B5)
      KORTP(B6+2)=TABLE(ICODEP,K1,B5+3)*IR3+TABLE(ICODEP,K1,B5+2)
      KORTP(B6+1)=TABLE(ICODEP,K1,B5+5)*IR3+TABLE(ICODEP,K1,B5+4)
C     FIND LOWEST ELEVATION STACK COEFFICIENTS
      WRITE(6,900) ICODEP,K1,B5
 900  FORMAT(10X,3I5)
      L=L+1
      K1=K-1
      B6=3
      IF(B8)1,1,2
  1   B8=1
      GO TO 10
  2   B6=0
C     INDICATES NORMAL LINEAR INTERPOLATION
      WRITE(6,800) KORTP
 800  FORMAT(10X,E15.8)
      STACK(1)=0
C     COMPUTE CROSSRANGE EFFECOT OF CORIOLIS KBYP
      STACK(6)=KORTP(3)
      STACK(7)=KORTP(6)
      CALL INTERP
      KBYP=PRTET
C     COMPUTE HORIZONTAL CORIOLIS EFFECT COEFFICIENT DEYP
      STACK(6)=KORTP(2)
      STACK(7)=KORTP(5)
      CALL INTERP
      KEYP=PRTET
C     COMPUTE VERTICAL CORIOLIS EFFECT COEFFICENT KGYP
      STACK(6)=KORTP(1)
      STACK(7)=KORTP(4)
      CALL INTERP
      KGYP=PRTET
      KBYP=KBYP*XNEWOK
      KEYP=KEYP*XNEWOK
      KGYP=KGYP*XNEWOK
C     COMPUTE CORIOLIS EFFECT IN SLANT RANGE FOR CONSTANT R3 KTR3
      TMP=R2J2*SINBY3*COSLAR
      KYP=KGYP-KEYP
      AL1=TMP*KYP
```

```
      AL2=AL1*SINE3
      AL3=AL2*COSE3
      KTR3=AL3*T2J2
C     COMPUTE VERTICAL ELEVATION RATE FCR CONSTANT TIME OF FLIGHT KTRRE3
      AL=SINE3**2
      AL1=KYP*AL
      AL2=KGYP-AL1
      AL3=AL2*TMP
      KTRRE3=AL3*T2J2/TANW
C     CCMPUTE CORIOLIS EFFECT ON TIME OF FLIGHT KT3
      KT3=KTRRE3*DT3DR
C     CCMPUTE CORIOLIS EFFECT ON ADVANCE RANGE R3 KR3
      KR3=KTRPE3+KTR3
C     COMPUTE SECOND PARTIAL SUM OF TIME OF FLIGHT EFFECTS SQUARED
      T2J2SQ=T2J2**2
C     CCMPUTE CROSSRANNGE CORIOLIS EFFECT KRLS
      GK=3.5748889
      TEMP=T2J2SQ*GK
      TANE4=SINE4/COSE4
      IF(TANE4-1000000)3,4,4
    4 TANE4=1000000
    3 AL=R2J2*TANE4
      WRITE(6,700) TANE4
  700 FCRMAT(10X,E15.8)
      WRITE(6,700) TEMP
       WRITE(6,700) SINE4
      WRITE(6,700) COSE4
      WRITE(6,700) AL
      AL1=AL*COSE3
      WRITE(6,700) AL1
      AL2=TEMP-AL1
      WRITE(6,700) AL2
      AL3=AL2*COSBY3
      WRITE(6,700) AL3
      TMP1=AL3*CCSLAR
      WRITE(6,700) TMP1
      WRITE(6,700) COSLAR
      WRITE(6,700) SINLAR
      AL4=R2J2*CCSE3
      WRITE(6,700) AL4
      AL5=AL4*SINLAR
      WRITE(6,700) AL5
      AL6=AL5+TMP1
      WRITE(6,700) AL6
      AL7=AL6*KBYP
      WRITE(6,700) AL7
      WRITE(6,700) KGYP
      WRITE(6,700) KEYP
```

```
      WRITE(6,700) KBYP
      KRLS=AL7*T2J2
      WRITE(6,100) KT3,KRLS,KR3
  100 FORMAT(10X,3E15.8)
      RETURN
      END

      SUBROUTINE DRIFT
C     THIS SUBROUTINE COMPUTES THE CROSSRANGE EFFECT OF PROJECTILE DRIFTOPS00120
C     THE BALLISTIC DRIFT EFFECTS ARE COMPUTED BY DRIFT BECAUSE THE     OPS00130
C     CURVE-FIT POLYNOMIAL FOR THE DRIFT EFFECT IS LINEAR AND DOES NOT  OPS00140
C     FIT THE NORMAL PATTERN THAT THE TABLE LOOKUP SUBROUTINES EXPECT   OPS00150
C     KORIND IS SET TO 2 TO INDICATE A NON-STANDARD TABLE. TWO POLY-    OPS00160
C     NOMIALS ARE NEEDED FOR THE AA MODE.                               OPS00170
C                                                                       OPS00180
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
      COMMON/CDR/DORLS
      COMMON/CCR/KRLS,KR3,KTRRE3,KT3,T2J2SQ,TANE4
      REAL*4 IR3
      INTEGER*4 B5
C
      TMP3=0
      KORIND=2
      ICODEP=11
      CALL TMONIT
CC    GET HIGEST STACK COEFFICIENTS FIRST
      L=1
      K1=K
      B8=0
C     ALLOWS FOR TWO PASSES THRU STACKS
C     LOCATION OF FIRST COEFFICIENT
   10 B5=HOLDB3(L)
      STACK(6+B8)=TABLE(ICODEP,K1,B5+1)*IR3+TABLE(ICODEP,K1,B5)
      WRITE(6,102) STACK(6+B8),IR3
      WRITE(6,700) TABLE(ICODEP,K1,B5+1),TABLE(ICODEP,K1,B5)
  700 FORMAT(10X,2E15.8)
      WRITE(6,701) ICODEP,K1,B5
  701 FORMAT(10X,3I5)
      L=L+1
      K1=K-1
      IF(B8)1,1,2
    1 B8=1
      GO TO 10
    2 B6=0
```

```
C       NORMAL LINEAR INTERPOLATION                                      OPS00190
        CALL INTERP
C       COMPUTE CROSSRANGE EFFECT ON PROJECTILE                          OPS00200
        DCRLS=PRTET*T2J2SQ
        WRITE(6,100) DORLS
  100   FORMAT(10X,'DORLS=',E15.8)
        WRITE(6,101) PRTET,T2J2SQ
  101   FORMAT(10X,2E15.8)
        WRITE(6,102) STACK(6),STACK(7)
  102    FORMAT(10X,2E15.8)
        RETURN
        END

        SUBROUTINE ZLIMIT                                                ZLI00010
C                                                                        OPS00210
C       THIS SUBROUTINE LIMITS THE Z COORDINATE OF A TARGET TO A MINIMUM OPS00220
C       BASED UPON TARGET HORIZONTAL RANGE,EARTH'S CURVATURE, HEIGHT OF  ZLI00030
C       GUN ABOVE THE SEA SURFACE                                        ZLI00040
C                                                                        ZLI00050
        COMMON/CZ/RHGEN,RZGEN                                            ZLI00060
C
        X=7.18E-08                                                       ZLI00070
        AL=RHGEN*X                                                       ZLI00080
        AL1=AL*RHGEN                                                     ZLI00090
        PVD=14.43
        RZS12=-AL-PVD-2                                                  ZLI00100
        Y=RZGEN-RZS12                                                    ZLI00110
        IF(Y)1,2,2                                                       ZLI00120
    2   RETURN                                                           ZLI00130
    1   RZGEN=RZS12+20                                                   ZLI00140
        RETURN                                                           ZLI00150
        END                                                              ZLI00160

        SUBROUTINE RTWT
C                                                                        OPS00230
C       THIS SUBROUTINE IS CALLED FROM THE BCOMPH BALLISTICS SEGMENT AND OPS00240
C       CALCULATES THE EFFECT ON TIME OF FLIGHT AND RANGE DUE TO PROJ-   OPS00250
C       ECTILE WEIGHT. A MINOR BUT OCCASIONALLY SIGNIFICANT EFFECT ON THE OPS00260
C       THE NORMAL BALLISTIC SOLUTION IS THE EFFECT  ON RANGE AND TOF    OPS00270
C       DUE TO A VARIATION IN PROJECTILE WEIGHT. THE EFFECT OF WEIGHT CAN OPS00280
C       BE CONSIDERED AS THE SUM OF EQUIVALENT EFFECTS OF INITIAL VELOCITYOPS00290
C       AND BALLISTIC AIR DENSITY. DUE TO INTERIOR BALLISTICS EFFECTS AN  OPS00300
C       INCREASE OF PROJECTILE WIGHT WILL PRODUCE A DECREASE IN DRAG ACC- OPS00310
C       ELERATION WHICH WOULD BE EQUIVALENT TO A DECREASE IN BALLISTIC AIROPS00320
C       DENSITY THUS THE RANGE AND TIME OF FLIGHT EFFECTS DUE TO A PRO-  OPS00330
C       JECTILE WEIGHT BARIATION CAN BE COMPUTED AS THE SUM OF THE EQUIVA-OPS00340
```

```
C     LENT EFFECTS ON INITIAL VELOCITY AND BALLISTIC AIR DENSITY          OPS00350
C                                                                         OPS00360
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UC
      COMMON/OF/HT3,DT3DH,DT3HDE,DT3HDR
      COMMON/OD/DT3DU,DT3UDR,DT3UDE,UT3,DT3DR
      COMMON/CE/DRDH,DRDHDE,HRH
      COMMON/CC/DRDU,DRDUDE,URH
      COMMON/OH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
C
      AMMOTR=0
C     COMPUTE CHANGE IN RANGE DUE TO CHANGE IN PROJECTILE WEIGHT WTR3
      TMP7=HO*DRDH
      KWU=C.29980469
      TMP71=UC*KWU
      WTR31=IWT*(-1/70)
      WTR3=(TMP71*DRDU+TMP7)*WTR31
C     COMPUTE CHANGE IN TIME OF FLIGHT DUE TO CHANGE IN PROJECTILE WIGHT WTT3
      TMP74=HO*DT3DH
      WTT3=(TMP71*DT3DU+TMP74)*WTR31
      RETURN
      END


      SUBROUTINE GUNMOT
C     THIS SUBROUTINE IS CALLED FROM BCCMPI IN THE BASIC CALCULATION      OPS00370
C     PART OF BALLISTICS. IT CALCULATES THE BALLISTICS EFFECTS DUE TO     OPS00380
C    GUN MOTION. SPECIFICALLY, THESE ARE THE EFFECTS ON ADVANCE RANGE     OPS00390
C     AND TIME OF FLIGHT AND THE DEFLECTION AND CROSSRANGE EFFECTS ON     OPS00400
C     GUN ORDERS. AS A BY-PRODUCT, THE GUN MOTION VECTOR INCREMENT TO     OPS00410
C     THE PROJECTILE VELOCITY IS OBTAINED UNDER CONDITIONS OF NO GUN      OPS00420
C     MOTION, THE PROJECTILE LEAVES THE MUZZLE WITH AN INITIAL VELOCITY OPS00430
C     VECTOR (U) AT INERTIAL TRUE BEARING BYG, AND INERTIAL ELEVATION     OPS00440
C     ANGLE EG WITH GUN MOTION, THE COMPONENT ALONG THE U VECTOR (DUMOG)OPS00450
C     WILL INCREASE THE MAGNITUDE OF U.                                   OPS00460
C                                                                         OPS00470
      COMMON/T120/IH,IPT,ITF,IU,IWT,HC,UC
      COMMON/CH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
      COMMON/OC/DRDU,DRDUDE,URH
      COMMON/CD/DT3DU,DT3UDP,DT3UDE,UT3
      COMMON/CA/T3,DT3DR,DT3DE,DT3RDE
      COMMON/ I/ANGLEW,DMR30G,DMBOG,SINEGB,COSEGB,TANEG,
     CTANW,TOVERW
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/OB/COSE4,DEGDE,DEGDR,EGB1,E4,SINE4,DEGRDE
      COMMON/CGU/DRDMEY,DRDMRY,DT3DMY,DT3MEY,MR3CR3,MR3OT3,MOLS,MORLS
C
```

```
C     COMPUTE U COMPONENT OF GUN MOTICN
      DMUOG=DMR30G*COSE4
      UMC=3*DMUCG+UO
C     COMPUTE DOWNRANGE COMPONENT OF IV WITH GUN MOTION
      R3UMC=UMO*COSEGB
      RH1UMO=R2J1/R3UMO
C     COMPUTE MOLS THE DEFLECTION EFFECT OF GUN MCTION CN GUN ORDERS
      MCLS=3*RH1UMO
C     CCMPUTE GUN MOTION EFFECTS CN CROSSRANGE MCRLS
      MORLS=MCLS*DMBOG
C     CCMPUTE PARTIAL ELEVATION CCMPONENT OF GUN MOTION WITH RANGE      OPS00480
C     (DEDR) AND ITS RECIPROCIAL DRDSEV                                 OPS00490
      DEDR=DEGDR*2*3.1416*UMO
      DRDSEV=1/DEDR
      DRDESE=DRDSEV*SINEGB
      DRDMRH=DRDU*COSEGB-DRDESE
C     CCMPUTE DERIVATIVE OF RANGE WITH RESPECT TC ELEVATION GUN VELCCITYOPS00500
      DRDMRY=3*DRDMRH
      DRDECE=DRDSEV*COSEGB
      DRDME3=DRDU*SINEGB+DRDECE
C     COMPUTE DERIVATIVE OF RANGE WITH RESPECT CT ELEVATICN GUN VELCCITY DRDMEY
      DRDMEY=3*DRDME3
C     COMPUTE THE DERIVATIVE OF SLANT RANGE WITH RESPECT TC VERTICAL    OPS00510
C     GUN VELOCITY DRDE3G                                               OPS00520
      DRDE3G=DRDMEY*COSE3+DRDMRY*SINE3
C     COMPUTE THE DERIVATIVE OF SLANT RANGE WITH RESPECT TC DOWN        OPS00530
C     HORIZONTAL GUN VELOCITY DRD30G                                    OPS00540
      DRD30G=DRDMRY*COSE3-DRDMEY*SINE3
C     COMPUTE DOWN RANGE EFFECT DUE TO PROJECTILE WEIGHT MR30R3         OPS00550
      MR30R3=DRD30G*DMR30G
      DT3DUC=DRDESE*DT3DR
      DT3DMR=DT3DU*COSEGB-DT3DUC
C     CCMPUTE EFFECT ON TIME OF FLIGHT DUE TO ELEVATION GUN             OPS00560
C     VELOCITY DT3MEY                                                   OPS00570
      DT3DMY=3*DT3DMR
      DT3DUS=DRDECE*DT3DR
      DT3DME=DT3DU*SINEGB+DT3DUS
C     CCMPUTE EFFECT ON TIME OF FLIGHT DUE TO ELEVATION GUN VELOCITY DT3MEY
      DT3MEY=3*DT3DME
C     CCMPUTE EFFECT ON TIME OF FLIGHT DUE TO HCRIZCNTAL GUN            OPS00580
C     VELCCITY DTM30G                                                   OPS00590
      DTM30G=DT3DMY*COSE3-DT3MEY*SINE3
C     CCMPUTE EFFECT ON TIME OF FLIGHT DUE TO DCWNRANGE GUN VELOCITY    OPS00600
      MR30T3=DTM30G*DMR30G
      RETURN
      END
```

```
      SUBROUTINE WIND
C                                                                       OPS00610
C     THIS SUBROUTINE CALCULATES WR3, THE TOTAL EFFECTS OF WIND ON      OPS00620
C     ADVANCE RANGE R3 AND WT3 THE EFFECT ON TIME OF FLIGHT T3, AND WRLSOPS00630
C     THE WIND EFFECT ON CROSSRANGE                                     OPS00640
C
      COMMON/OGU/DRDMEY,DRDMRY,DT3DMY,DT3MEY,MR30R3,MR30T3,MOLS,MORLS
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/CH/TFT3,DT3FDE,J1R3,R2J1,WTR3,WTT3,T2J1,DT2DR
     C,DT3DTF
      COMMON/ I/ANGLEW,DMR30G,DMBOG,SINEGB,COSEGB,TANEG,
     CTANW,TOVERW
      COMMON/T13/WX,WY,WXR,WYR
      COMMON/CW/DRDWRH,DTDWRH,WBG,WRHG,WRLS,WR3,WT3
C     COMPUTE DOWNRANGE COMPONENT OF WIND
      WRHG=WXR*SINBY3+WYR*COSBY3
C     COMPUTE CROSSRANGE COMPONENT OF WIND
      WBG=WXR*COSBY3-WYR*SINBY3
C     COMPUTE CROSSRANGE EFFECT
      WRLS=(T2J1-MOLS)*WBG
C     COMPUTE PARTIAL OF SLANT RANGE WITH RESPECT TO ELEVATION RANGE    OPS00650
C     WIND DRDWE3                                                       OPS00660
      DRDWE3=TOVERW-DRDMEY*3
C     COMPUTE PARTIAL OF SLANT RANGE WITH RESPECT TO SLANT RANGE WIND DRWR3
      DRDWR3=T2J1-DRDMEY
C     COMPUTE DERIVATIVE OF SLANT RANGE WITH RESPECT TO HORIZONTALDOWN  OPS00670
C     RANGE WIND DRDWRH                                                 OPS00680
      DRDWOH=DRDWE3*SINE3
      DRDWRH=DRDWR3*COSE3-DRDWOH
C     COMPUTE WIND EFFECT ON R3 WR3
      WR3=DRDWRH*WRHG
C     COMPUTE PARTIAL OF SLANT RANGE WITH RESPECT TO VERTICAL RANGE WIND
C     DRDWVG
      DRDWVG=DRDWE3*COSE3+DRDWR3*SINE3
C     COMPUTE DERIVATIVE OF TIME OF FLIGHT WITH RESPECT TO SLANT RANGE
C     WIND DTDWR3
      DTDWR3=-DT3DMY
C     COMPUTE DERIVATIVE OF TIME OF FLIGHT WITH RESPECT TO HORIZONTAL   OPS00690
C     DOWNRANGE WIND DTDWRH
C     COMPUTE DERIVATIVE OF TIME OF FLIGHT WITH RESPECT TO VERTICAL     OPS00700
C     RANGE                                                             OPS00710
      DTDWRH=DTDWR3*COSE3
C     COMPUTE DERIVATIVE OF TIME OF FLIGHT WITH RESPECT TO ELEVATION RANGE
C     WIND DT3DWE
      DT3WDE=TOVERW*DT2DR-DT3MEY
      DTDWRH=DTDWRH-DT3DWE*SINE3
C     COMPUTE WIND EFFECT ON TIME OF FLIGHT WT3
      WT3=DTDWRH*WRHG
```

```
      DT3DWV=DT3DWE*COSE3+DTDWR3*SINE3
      RETURN
      END

      SUBRCUTINE TMONIT                                                 TMC00C10
C     FCR A SINGLE PARAMETER THIS SUBROUTINE DOES INITIAL SETUP AND
C     LCCATES THE PROPER TABLE CORRECPONDING TC TARGET ELEVATION.       OPS00730
C     THEN CALL CTHER SUBROUTINES TO OBTAINE VALUS CF PARAMETER FROM    OPS00740
C     TABLE, CHECKS FOR RANGE VIOLATICN AND HIGH CR LOW ELEVATION ONCE  OPSC0750
C     FOR EACH PASS THRU BALLISTICS SECTION.                            OPS00760
C                                                                       OPSC0770
      COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
      COMMON/T22/E3,SINE3,COSE3,BY3,SINBY3,COSBY3
      COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      COMMCN/T01/RCOEFF(8),KMTABL(11,2),ETABL1(17,11),PUSHTB(11,2)
      INTEGER*4 T17,PUSHTB
      INTEGER*4 ETABL1
C                                                                       OPS00720
      IF(ICODEP-2)1,2,2                                                 TM000090
C     CHECK IF RANGE GREATER THAN 50 YARDS                              OPS00780
    1 IF(RTEMP-50)3,3,4                                                 TMC00100
    3 LIMVIO=-1                                                         TMC00110
C     MINIMUM RANGE LIMIT VIOLATED
C     CHECK FOR E3 GREATER THAN OR LESS THAN 35 DEGREES
    4 MIDIND=0                                                          TM000120
      IF(35-E3)5,5,6                                                    TM000140
    5 MIDIND=4                                                          TMC00150
      IF(E3-80)6,8,8                                                    TM000160
    8 E3D=80                                                            TMC00170
      GC TO 9                                                           TMC00180
    6 E3D=E3                                                            TMC0C190
    9 STACK(2)=E3D                                                      TM000200
      LR3=RCOEFF(MIDIND+1)+E3D*RCOEFF(MIDIND+2)+E3D**2*RCOEFF(MIDIND+3) TMOC0210
     C+E3D**3*RCCEFF(MIDIND+4)                                          TM000220
      IF(LR3-RTEMP)10,10,2                                              TMC00230
   10    R3=LR3                                                         TMC00240
      LIMVIO=1                                                          TM000250
C     MAXIMUM RANGE LIMIT VIOLATED
C     INITIALIZSATICN FOR EACH LOOKUP
    2 RHCLD=RTEMP                                                       TMC00260
      EXACTE=0                                                          TMC00270
      IF (ICODEP-3)30,31,31                                             GAR00250
   30    STACK(1)=1                                                     GAR00260
C     PARABOLIC INTERPOLATION
```

```
      GO TO 33                                                          GAR00270
   31 STACK(1)=0                                                        GAR00280
   33 B5ADD1=STACK(1)                                                   GAR00290
C     FIND THE NUMBER OF TABLE LOOKUPS
      ISRCH=B5ADD1+1                                                    TMO00310
C     FIND THE TOTAL NUMBER OF ELEVATICN STACKS FCR THIS PARAMTERER
      BIGKM5=KMTABL(ICODEP,1)                                           TMC00320
      SETBK=KMTABL(ICODEP,2)                                            TMO00330
      B5ADD2=SETBK*MIDIND                                               TMC00340
C     HIGH OR LOW ELEVATICN MODE INDICATOR 0 LOW GREATER THAN 0 HIGH
      IF(B5ADD2)12,12,13                                                TMC00350
   12 K=PUSHTB(ICODEP,2)                                                TMO00360
C      FIRST ELEVATION STACK FOR LOW AA MODE
      GO TO 14                                                          TMO00370
   13 B5ADD2=B5ADD2-1                                                   TMC00380
      K=PUSHTB(ICODEP,1)                                                TMO00390
C     FORST ELEVATION STACK FCR HIGH AA MODE
   14 CALL DECSI                                                        TMC00400
C     IF HIGH AA GET NEXT HIGHER STACK NCW
      IF(B5ADD2.GT.0) CALL FINDAL                                       TMO00410
C     CHECK FCR EXACTER ELEVATION STACK AT INITIAL VALUE
      IF(STACK(2)-STACK(4).EQ.0) EXACTE=2                               TMO00420
      IF(B5ADD2)15,15,16                                                TMC00430
   16 B5ADD2=B5ADD2-1                                                   TMO00440
      B6=K+2                                                            TMC00450
      GO TO 17                                                          TMO00460
   15 B6=K+1                                                            TMO00470
C     CHECK IF STACK ELEVATICN = ORGREATER THAN E3 IF EQUAL SET
C     EXACTE =1
   17 IF(STACK(2)-STACK(3))18,19,20                                     TMO00480
   19 EXACTE=1                                                          TMC00490
C     CHECK IF WE WANT CNE OR TWO STACKS GREATER THAN E3
  18  IF(B5ADD1)21,21,22                                                TMO00500
   22 B5ADD1=B5ADD1-1                                                   TMC00510
C     IF THERE IS AN EXACTE STACK KEEP TRACK OF IT
      IF(EXACTE)20,20,23                                                TMC00520
   23 EXACTE=EXACTE+1                                                   TMC00530
C     CHECK IF MAX RANGE IS GREATER THAN R3
   20 IF(ETABL1(B6,ICODEP)-RHOLD)21,21,24                               TMO00540
C     GET NEXT STACK AND GO THRU LOOP AGAIN
   24 CALL FINDAL                                                       TMO00550
C     CHECK IF ANOTHER STACK IS ABAILABLE
      IF(B6-BIGKM5)25,21,21
   25 B6=B6+1                                                           TMC00570
      GO TO 17                                                          TMC00580
   21 CALL AASRCH
      RETURN                                                            TMC00600
      END                                                               TMC00610
```

```
      SUBROUTINE AASRCH                                                 AAS00010
C     THIS SUBROUTINE CONTROLS THE SELECTION OF COEFFICIENTS FOR BRACKETOPS00800
C     ING ELEVATION STACKS,THE CUBIC POLYNOMIAL EVALUATION, AND THE     OPS00810
C     INTERPOLATION BETWEEN ELEVATION STACKS FOR AA BALLISTICS. THIS    OPS00820
C     SUBROUTINE ALSO DETERMINES THE APPROPRIATE DERIVATIVES NEEDED FOR OPS00830
C     AA BALLISTICS.                                                    OPS00840
C                                                                       OPS00850
      COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
      INTEGER*4 T17                                                     GAR00340
C     B6= THE NUMBER OF TABLE LOOKUPS
C                                                                       OPS00790
      B6=ISRCH                                                          AAS00050
C     CHECK FOR STANDARD TABLES OF 4 COEFFICIENTS
      IF(KORIND-4)2,1,2
C     NON-STANDARD TABLES LOOKUP
    2 CALL SEARCH                                                       AAS00070
      IX=B6+1
C     ADDRESS OF FIRST COEFFICIENT IS STORED IN HOLDB3 MATRIX, HIGHEST  OPS00860
C     STACK IN HOLDB3(1)                                                OPS00870
      HOLDB3(IX)=T17
      IF (B6)20,20,21
   21 B6=B6-1
      GO TO 2
   20 RETURN
C     STANDARD TABLE LOOKUP
C     FIND ADDRESS OF FIRST COEFFICIENT AND EVALUATE POLYNOMIAL,EVAL-   OPS00880
C     UATED PARAMETERS ARE STORED IN STACK MATRIX, HIGHEST ELEVATION IN OPS00890
C     STACK(6) HIGHEST RANGE DERIVATIVE IN STACK (9)                    OPS00900
    1    CALL SEARCH                                                    AAS00130
      CALL POLYN                                                        AAS00140
      STACK(6+B6)=TEMP2                                                 AAS00150
      STACK(9+B6)=TMP10                                                 AAS00160
      IF (B6)4,4,5                                                      AAS00170
    5 B6=B6-1                                                           AAS00180
      GO TO 1                                                           AAS00190
    4 CALL INTERP
C     GET INTERPOLATED VALUE OF PARAMETERS AND STORE IN TEMP2
      TEMP2=PRTET
C     GER ELEVATION DERIVATIVE AND STORE IN DPDEP
      DPDEP=DPDE                                                        AAS00210
C     NORMAL LINEAR INTERPOLATION
      B6=0                                                              AAS00220
C     CHECK IF WE WANT A DERIVATIVE WITH RESPECT TO RANGE
      IF (TMP3)6,6,7                                                    AAS00230
```

```
C       NO RANGE DERIVATIVE
    6   TMP10=TMP3                                                              AAS00240
        RETURN                                                                  AAS00250
C       WANT RANGE DERIVATIVE
C       CHECK FOR LINEAR OR PARABOLIC INTERPOLATION
    7   IF(STACK(1))8,8,9                                                       AAS00260
C       IF MIDDLE ELEVATION STACK IS GREATER THAN ADVANCE ELEVATION SET         OPS00910
C       B6=1 FOR SPECIAL AA DERIVATIVE INTERPOLATION                            OPS00920
    9   IF(STACK(4)-STACK(2))8,8,10                                             AAS00270
   10   B6=1                                                                    AAS00280
    8   STACK(6+B6)=STACK(9+B6)                                                 AAS00290
        STACK(7+B6)=STACK(10+B6)                                                AAS00300
        STACK(1)=0                                                              AAS00310
        EXACTE=0                                                                AAS00320
        CALL INTERP                                                             AAS00330
        TMP10=PRTET                                                             AAS00340
C       MIXED PARTIAL OF ELEVATION AND RANGE STORED IN DPDED                    OPS00930
        DPDED=DPDE                                                              AAS00350
        RETURN                                                                  AAS00360
        END                                                                     AAS00370


        SUBROUTINE SEARCH                                                       SEA00010
C       THIS SUBROUTINE FINDS THE DIFFERENCE IN RANGE (IR3) TO BE USED IN       OPS00950
C       DETERMING EACH PARAMETER AND LOCATION OF FIRST COEFFICIENT              OPS00960
C       TO BE USED (T17)                                                        OPS00970
        COMMON/T21/RTEMP,R3,RX3,RY3,RHOLD
        COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
        COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
       CDPDE,ISRCH
        COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
        REAL*4 IR3
        INTEGER*4 T17                                                           GAR00340
C                                                                               OPS00940
C       WANT TO START AT THE LOWEST STACK FIRST SO NUMBER OF LOOKUPS IS         OPS00980
C       SUBTRACTED FROM K THE HIGHEST STACK NUMBER                              OPS00990
        K1=K-B6                                                                 GAR00010
C       FIND THE NUMBER OF RANGE PARTITIONS
        N=TABLE(ICODEP,K1,2)
C       FIND THE LOCATION OF HIGHEST RANGE PARTITION
        N=N+2                                                                   SEA00040
C       FIND THE FIRST RANGE PARTITION LESS THAN R3
    5   AR3=RHOLD-TABLE(ICODEP,K1,N)                                            GAR00090
        IF(AR3)1,1,2                                                            SEA00060
    2   T16=N-3                                                                 SEA00070
C       T16=THE NUMBER OF THE RANGE PARTITION WHICH IS LESS THAN R3             OPS01000
C       MINUS 1                                                                 OPS01010
        IR3=AR3                                                                 SEA00080
```

```
      GC TO 6                                                           SEA00090
C     N=2 CORRESPONDS TC ZERO RANGE                                     OPS01020
 1    IF(N-2)3,3,4                                                      SEA00100
 4    N=N-1                                                             SEA00110
      GC TO 5                                                           SEA00120
 3    IR3=RHOLD                                                         SEA00130
      AK=0                                                              SEA00140
C     AK= THE NUMBER OF CCEFFICIENTS TC BYPASS                          OPS01030
      GO TO 7                                                           SEA00150
 6    AK=KORIND*T16                                                     SEA00160
C     FIND THE LOCATION OF FIRST COEFFICIENT
  7   T17=AK+3+TABLE(ICODEP,K1,2)                                       GAR00100
      WRITE(6,102)T17                                                   GAR00060
102   FCRMAT(10X,'T17=',I4)
      WRITE(6,100)K1                                                    GAR00020
100   FCRMAT(10X,'K1=',I3)                                              GAR00030
      WRITE(6,101)IR3                                                   GAR00040
101   FCRMAT(10X,'IR3=',E15.8)                                          GAR00050
      RETURN                                                            SEA00180
      END                                                               SEA00190


      SUBROUTINE POLYN                                                  POL00010
C     THIS SUBROUTINE EFALUATES THE PCLYNOMIAL FRCM THE COEFFICIENTS ANDOPS01050
C     FINDS THE DERIVATIVE WITH RESPECT TO RANGE. THE PARAMETER IS FOUND
C     BY P=IR3**3*C3+IR3**2*C2+IR3*C1+C0
      COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
      CCMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      CCMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
      REAL*4 IR3
      INTEGER*4 T17                                                     GAR00340
C                                                                       OPS01040
C     WANT TO START AT THE LCWEST EVEVATICN STACK FIRST SO THE NUMBER OF
C      LOOKUPS IS SUBTRACTED FROM K
      K1=K-B6
C     TEMP2 HOLDS THE EVALUATED PCLYNOMIAL
      TEMP2=IR3**3*TABLE(ICODEP,K1,T17+3)+IR3**2*TABLE(ICODEP,K1,T17+2)+GAR00110
     CIR3*TABLE(ICODEP,K1,T17+1)+TABLE(ICODEP,K1,T17)                   GAR00120
      TMP10=0                                                           POL00050
C     CHECK IF WE WANT A DERIVATIVE WITH RESPECT TC RANGE TMP3=0        OPS01060
C     NC, TMP3=1 YES                                                    OPS01070
      IF(TMP3-1)1,2,2                                                   POL00060
 1    WRITE(6,102)TEMP2
102   FCRMAT(10X,'TEMP2=',E15.8)
      RETURN
C     COMPUTE DERIVATIVE BY FORMULA TMP10= 3**IR3**2*C3+2*IR3*C2+C1
 2    TMP10=3*IR3**2*TABLE(ICODEP,K1,T17+3)+2*IR3*TABLE(ICODEP,K1,T17+2)GAR00130
```

```
      C+TABLE(ICODEP,K1,T17+1)                                          GAR00140
       WRITE(6,100)TEMP2                                                GAR00150
 100   FORMAT(10X,'TEMP2=',E15.8)                                       GAR00160
       WRITE(6,101)TMP10                                                GAR00170
 101   FORMAT(10X,'TMP=',E15.8)                                         GAR00180
       RETURN                                                           POL00100
       END                                                              POL00110


       SUBROUTINE SLIDE                                                 SLI00010
C     THIS SUBROUTINE SLIDES THE LOCATION OF PARAMETER VALUES AND       OPS01080
C      DERIVATIVES VALUES UP BY 1 AT A TIME IN STACK                    OPS01090
       COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
      CDPDE,ISRCH
       INTEGER*4 T17                                                    GAR00340
C                                                                       OPS01100
       DO 1 I=3,10
       STACK(14-I)=STACK(13-I)
   1   CONTINUE                                                         SLI00050
       RETURN                                                           SLI00060
       END                                                              SLI00070


       SUBROUTINE FINDAL                                                FIN00010
C       THIS SUBROUTINE FINDS THE NEXT HIGHER ELEVATION STACK BY
C      INCREMENTING K
       COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
       INTEGER*4 T17                                                    GAR00340
       CALL SLIDE                                                       FIN00030
       K=K+1                                                            FIN00040
       CALL DECSI                                                       FIN00050
       WRITE(6,100)K                                                    GAR00230
 100   FORMAT(10X,'K=',I3)                                              GAR00240
       RETURN                                                           FIN00060
       END                                                              FIN00070


       SUBROUTINE DECSI                                                 DEC00010
C      THIS SUBROUTINE DECODES THE ELEVATION INFORMATION FROM EACH STACK CPS01110
C      AND STORES IT IN STACK (3)                                       OPS01120
       COMMON/A2/TABLE(11,17,32),K,T17,LIMVIO
       COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
      CDPDE,ISRCH
       COMMON/N1/ICODEP,DPDEP,DPDED,TEMP2,TMP3,TMP10
C                                                                       OPS01130
       STACK(3)=TABLE(ICODEP,K,1)                                       DEC00030
       WRITE(6,100)STACK(3)
 100   FORMAT(10X,'STACK(3)=',E15.8)
```

```
      RETURN                                                            DEC00040
      END                                                               DEC00050

      SUBROUTINE INTERP                                                 INT00010
C     THIS SUBROUTINE PERFORMS LINEAR OR PARABOLIC INTERPOLATIONS ON    OPS01140
C     INPUT PARAMETERS WHICH ARE FUNCTIONS OF ADVANCED RANGE AND ADVANCEOPS01150
C     ELEVATION. THESE TERMS ARE POLYNOMIAL EVALUATED AT THE ADVANCED   OPS01160
C     RANGE AND CERTAIN ELEVATIONS THE PARAMETER INTERPOLATIONS ARE     OPS01170
C     PERFORMED TO OBTAIN THE VALUE AT THE ACTUAL ADVANCED ELEVATION.   OPS01180
C     THE INPUTS TO INTERP ARE PROVIDED FOR 2 OR 3 CHOSEN ELEVATION     OPS01190
C     VALUES OR STACKS BRACKETING THE ADVANCED ELEVATION AND ARE INSER- OPS01200
C     TED THROUGH THE STACK MATRIX ORIENTED FROM HIGHEST TO LOWEST      OPS01210
C     ELEVATION                                                         OPS01220
      COMMON/A1/KORIND,STACK(11),HOLDB3(3),PRTET,IR3,EXACTE,B6,
     CDPDE,ISRCH
      INTEGER*4 T17                                                     GAR00340
C     CHECK FOR LINEAR OR PARABOLIC INTERPOLATION
      IF (STACK(1)-1)1,2,2                                              INT00030
C     LINEAR INTERPOLATION B6=0 NORMAL LINEAR,B6=1 SPECIAL FOR SOME     OPS01230
C     AA DERIVATIVES                                                    OPS01240
C     P2-P1
    1 P1=STACK(6+B6)-STACK(7+B6)                                        INT00040
C     E2-E1
      DE1=STACK(3+B6)-STACK(4+B6)                                       INT00050
C     COMPUTE DERIVATIVE WITH RESPECT OT ELEVATION DPDE
      DPDE=P1/DE1                                                       INT00060
C     E3C-D1
      DE2=STACK(2)-STACK(4+B6)                                          INT00070
C     COMPUTE INTERPOLATED VALUE PRTET
      PRTET=DPDE*DE2+STACK(7+B6)                                        INT00080
      RETURN                                                            INT00090
C     PARABOLIC INTERPOLATION
C     P2-P1
    2 DP1=STACK(7)-STACK(8)                                             INT00100
C     E2-E1
      DE3=STACK(4)-STACK(5)                                             INT00110
C     COMPUTE ELEAVATION DERIVATIVE AT LOWEST ELEVATION INTERVAL DPDE1H
      DPDE1H=DP1/DE3                                                    INT00120
C     P3-P2
      DP2=STACK(6)-STACK(7)                                             INT00130
C     E3-E2
      DE4=STACK(3)-STACK(4)                                             INT00140
C     COMPUTE ELEVATION DERIVATIVE AT HIGH ELEVATION INTERVAL DPDE3H    OPS01250
      DPD3H=DP2/DE4                                                     INT00150
C     COMPUTE FINITE DIFFERENCE SECOND DERIVATIVE OVER BOTH             OPS01260
C     INTERVALS DDPDE                                                   OPS01270
      DDPDH=DPD3H-DPDE1H                                                INT00160
```

```
C       E3-E1
        DE7=STACK(3)-STACK(5)                                          INT00170
        DDPDE=DDPDH/DE7                                                INT00180
C       COMPUTE FINITE DIFFERENCE DERIVATIVE AT LOWEST STACK DPDEK
C       E3D-E1
        DE5=STACK(4)-STACK(5)                                          INT00190
        DDE1=DE5*DDPDE                                                 INT00200
        DPDEK=DPDE1H-DDE1
        DE6=STACK(2)-STACK(5)                                          INT00220
        TMP17=DDPDE*DE6*2
C       COMPUTE ELEVATION DERIVATIVE DPDE
        DPDE=TMP17+DPDEK                                               INT00240
C       IF THERE IS AN EXACTE ELEVATION STACK USE ITS PARAMETER VALUE
        B7=EXACTE                                                      INT00250
        IF(B7)3,3,4                                                    INT00260
C       COMPUTE INTERPOLATED VALUE PRTET
  3     PRTET=((0.5*TMP17+DPDEK)*DE6)+STACK(8)
        RETURN                                                         INT00280
  4     PRTET=STACK(6+B7)                                              INT00290
        WRITE(6,200)B6                                                 GAR00310
 200    FORMAT(10X,'B6=',I3)                                           GAR00320
        RETURN                                                         INT00300
        END                                                            INT00310


        SUBROUTINE ARTAN
C
C       THIS SUBROUTINE FINDS THE INVERSE TANGENT, GIVEN TWO PARAMETERS    OPS01290
C       AND DETERMINES THE CORRECT QUADRANT                            OPS01300
C
        COMMON/RT/COS1,SIN1,TAN
C
C       CHECK FOR ZERO DENOMINATOR
        IF(ABS(COS1)-1.5E-5)10,10,11
 11     PHI=ATAN(SIN1/COS1)
        IF (SIN1.LT.0.0) GO TO 1
        IF (COS1.LT.0.0) GO TO 2
C
C       GET HERE FOR FIRST QUADRANT ANGLE                              OPS01310
        TAN=PHI*57.293
        GO TO 1000
C
C       GE HERE FOR SIN1 NEGATIVE
  1     IF (COS1.LT.0.0) GO TO 3
C
C       GET HERE FOR SECOND QUADRANT ANGLE                             OPS01320
        PHI=ABS(PHI)+3.1415927/2
        TAN=PHI*57.293
```

```
      GC TO 1000
C
C     GET HERE FOR FOURTH QUADRANT ANGLE                                OPS01330
  2   PHI=ABS(PHI)+3*3.1415927/2
      TAN=PHI*57.293
      GC TO 1C00
C
C     GET HERE FOR THIRD QUADRANT ANGLE
  3   PHI=PHI+3.1415927
      TAN=PHI*57.293
      GC TO 1C00
C
C     GET HERE WHEN THE ABSOLUTE VALUE CD COS1 IS LESS THAN 0.000015    OPS01340
   1C  TAN=90
      IF(SIN1)12,12,13
  12  TAN=270.0
      GC TO 1C00
   13 TAN=TAN
 1CC0 RETURN
      END
```

FIRST BALLISTIC TABLE

0.25000000E 01	0.50000000E 01	0.10000000E 02
0.50000000E 01	0.40000000E 01	0.40000000E 01
0.0	0.0	0.0
0.81000000E 04	0.76000000E 04	0.81000000E 04
0.12800000E 05	0.12900000E 05	0.13000000E 05
0.17900000E 05	0.16700000E 05	0.17100000E 05
0.20100000E 05	-0.34974755E-02	-0.41918755E-02
-0.38096944E-02	0.12058103E-02	0.12079733E-02
0.12069279E-02	0.32674269E-07	0.32762074E-07
0.31923889E-07	0.16752537E-11	0.16956610E-11
0.17281445E-11	0.11789101E 02	0.12832829E 02
0.12791668E 02	0.19907146E-02	0.20769057E-02
0.20690537E-02	0.78480014E-07	0.79382369E-07
0.82826546E-07	0.22032957E-11	0.27101221E-11
0.19844543E-11	0.24863998E 02	0.25228867E 02
0.24542694E 02	0.29904984E-02	0.30408958E-02
0.29724282E-02	0.61848766E-07	0.71283523E-07
0.51397571E-07	0.28494516E-11	0.39738976E-11
0.32850805E-11	0.37271606E 02	0.39167862E 02
0.41479630E 02	0.36103753E-02	0.38892061E-02
0.37697789E-02	0.69847715E-07	0.86932687E-07
0.10661853E-06	0.19360208E-10	0.38425110E-10
0.20725921E-10	0.0	0.0
0.50509995E 02	0.0	0.0
0.44667125E-02	0.0	0.0
-0.44618109E-09	0.0	0.0
0.33343110E-08	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

FIRST BALLISTIC TABLE

0.17500000E 02	0.25000000E 02	0.32500000E 02
0.50000000E 01	0.50000000E 01	0.50000000E 01
0.0	0.0	0.0
0.81000000E 04	0.81000000E 04	0.81000000E 04
0.13200000E 05	0.13700000E 05	0.13600000E 05
0.16000000E 05	0.16700000E 05	0.16200000E 05
0.18300000E 05	0.18500000E 05	0.17300000E 05
-0.29560241E-02	-0.40703043E-02	-0.42205453E-02
0.12059226E-02	0.12073196E-02	0.12070343E-02
0.34487030E-07	0.35200095E-07	0.36412043E-07
0.16065482E-11	0.16057173E-11	0.15721824E-11
0.12882056E 02	0.12938739E 02	0.12998223E 02
0.20932737E-02	0.21059385E-02	-0.21295245E-02
0.75142850E-07	0.76349011E-07	0.73353135E-07
0.37794646E-11	0.43286815E-11	0.57454918E-11
0.26008972E 02	0.27875519E 02	0.27874924E 02
0.31318283E-02	0.33538149E-02	0.34593889E-02
0.97963493E-07	0.99532429E-07	0.11563230E-06
0.42486067E-11	0.18975294E-10	0.30372982E-10
0.35642242E 02	0.39358093E 02	0.38190216E 02
0.38176882E-02	0.45796409E-02	0.47449805E-02
0.12642954E-06	0.15239954E-06	0.32450924E-06
0.33356526E-10	0.16996594E-09	0.17871009E-09
0.45498886E 02	0.49089996E 02	0.44038147E 02
0.49869716E-02	0.73166266E-02	0.62974133E-02
0.33200217E-06	-0.39996248E-05	0.31993761E-06
0.19036456E-09	0.18332479E-07	0.12781609E-08
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

FIRST BALLISTIC TABLE

0.40000000E 02	0.45000000E 02	0.50000000E 02
0.50000000E 01	0.60000000E 01	0.60000000E 01
0.0	0.0	0.0
0.81000000E 04	0.77000000E 04	0.65000000E 04
0.14300000E 05	0.12100000E 05	0.11400000E 05
0.16400000E 05	0.15000000E 05	0.13900000E 05
0.17200000E 05	0.16400000E 05	0.15800000E 05
-0.49694702E-02	0.17000000E 05	0.16700000E 05
0.12079945E-02	-0.52835867E-02	-0.39421432E-02
0.36857820E-07	0.12077424E-02	0.12045556E-02
0.16209768E-11	0.37571279E-07	0.39595079E-07
0.13060121E 02	0.16040468E-11	0.14275551E-11
0.21471460E-02	0.12258004E 02	0.98921366E 01
0.75212199E-07	0.20886378E-02	0.19173657E-02
0.66804912E-11	0.73567833E-07	0.62417541E-07
0.30857330E 02	0.65378172E-11	0.55467107E-11
0.39206222E-02	0.23429688E 02	0.21439667E 02
0.15915901E-06	0.31640455E-02	0.29692831E-02
0.84564272E-10	0.11295219E-06	0.13460090E-06
0.40578979E 02	0.26688915E-10	0.16096638E-10
0.58729537E-02	0.34208740E 02	0.29956024E 02
0.51118286E-06	0.45818202E-02	0.39915740E-02
0.77441120E-09	0.29451155E-06	0.21310615E-06
0.45998566E 02	0.20320133E-09	0.10570102E-09
0.84047653E-02	0.41760468E 02	0.39037048E 02
0.85710417E-06	0.67654736E-02	0.62141754E-02
0.66667205E-08	0.91666840E-06	0.33975215E-06
0.0	0.16666648E-08	0.13616877E-08
0.0	0.46509995E 02	0.45900024E 02
0.0	0.98000355E-02	0.14461782E-01
0.0	0.39996394E-05	-0.54453228E-04
0.0	0.10000807E-07	0.20822944E-06

FIRST BALLISTIC TABLE

0.55000000E 02	0.60000000E 02	0.65000000E 02
0.60000000E 01	0.60000000E 01	0.60000000E 01
0.0	0.0	0.0
0.72000000E 04	0.67000000E 04	0.70000000E 04
0.11500000E 05	0.10400000E 05	0.11600000E 05
0.14100000E 05	0.13400000E 05	0.14300000E 05
0.15700000E 05	0.15400000E 05	0.15400000E 05
0.16400000E 05	0.16200000E 05	0.16000000E 05
-0.55188315E-02	-0.43662786E-02	0.12069531E-02
0.12083359E-02	0.12050325E-02	-0.49820393E-02
0.38502005E-07	0.40173283E-07	0.39765769E-07
0.16012643E-11	0.14782073E-11	0.15681120E-11
0.11291344E 02	0.10322681E 02	0.10930743E 02
0.20298087E-02	0.19529739E-02	0.20218999E-02
0.70850206E-07	0.72393732E-07	0.65509482E-07
0.66418190E-11	0.51411896E-11	0.78941255E-11
0.21857910E 02	0.18802277E 02	0.22388031E 02
0.30537299E-02	0.27169837E-02	0.31799206E-02
0.13038471E-06	0.13824149E-06	0.12208204E-06
0.23905405E-10	0.12156998E-10	0.41806447E-10
0.31095810E 02	0.28526825E 02	0.32688782E 02
0.43398701E-02	0.39764419E-02	0.48650429E-02
0.19040306E-06	0.16822884E-06	0.41042381E-06
0.19736818E-09	0.13408284E-09	0.33022429E-09
0.39339233E 02	0.38227463E 02	0.38980469E 02
0.66321380E-02	0.65979213E-02	0.71484111E-02
0.72187152E-06	0.28173861E-06	0.10000003E-05
0.18181721E-08	0.20875475E-08	0.27777725E-08
0.44959991E 02	0.44759995E 02	0.44229996E 02
0.10916717E-01	0.11799712E-01	0.12432531E-01
-0.48341775E-09	-0.19972540E-05	-0.59923159E-05
0.28334391E-07	0.49993904E-07	0.66649591E-07

FIRST BALLISTIC TABLE

0.70000000E 02	0.75000000E 02	0.80000000E 02
0.60000000E 01	0.60000000E 01	0.60000000E 01
0.0	0.0	0.0
0.64000000E 04	0.69000000E 04	0.72000000E 04
0.11100000E 05	0.10800000E 05	0.10800000E 05
0.13600000E 05	0.13000000E 05	0.13100000E 05
0.15000000E 05	0.14800000E 05	0.14900000E 05
0.15800000E 05	0.15700000E 05	0.15600000E 05
-0.20820610E-02	-0.33271667E-02	-0.33796197E-02
0.12042467E-02	0.12068197E-02	0.12071456E-02
0.40758277E-07	0.39827619E-07	0.39681169E-07
0.15065128E-11	0.16479977E-11	0.16992640E-11
0.97699099E 01	0.10764632E 02	0.11381729E 02
0.19324361E-02	0.20064218E-02	0.20611000E-02
0.62336255E-07	0.73499336E-07	0.77397431E-07
0.68052896E-11	0.67833898E-11	0.73805797E-11
0.20937790E 02	0.20111404E 02	0.20151810E 02
0.30146658E-02	0.29214357E-02	0.29439295E-02
0.14150254E-06	0.15559669E-06	0.14778215E-06
0.26913929E-10	0.18597290E-10	0.23287483E-10
0.29781647E 02	0.27490082E 02	0.27986465E 02
0.43069944E-02	0.39403625E-02	0.40876716E-02
0.30139967E-06	0.22125425E-06	0.19263251E-06
0.18706996E-09	0.13275185E-09	0.17903551E-09
0.36915344E 02	0.36075699E 02	0.37016357E 02
0.64515248E-02	0.63371398E-02	0.68099126E-02
0.52858027E-06	0.18880974E-06	0.63638583E-06
0.18350153E-08	0.18803430E-08	0.24494740E-08
0.43360001E 02	0.43309998E 02	0.42939987E 02
0.11699837E-01	0.12549311E-01	0.12483109E-01
-0.14984453E-05	-0.69934404E-05	-0.69978569E-05
0.44996519E-07	0.74985394E-07	0.71661873E-07

SECOND BALLISTIC TABLE

0.25000000E 01	0.50000000E 01	0.10000000E 02
0.50000000E 01	0.40000000E 01	0.50000000E 01
0.0	0.0	0.0
0.65000000E 04	0.66000000E 04	0.66000000E 04
0.12100000E 05	0.12100000E 05	0.11800000E 05
0.16600000E 05	0.16600000E 05	0.16400000E 05
0.20100000E 05	-0.40924110E-01	0.18500000E 05
-0.39640784E-01	0.26602026E-01	-0.36135794E-01
0.26675086E-01	0.93329294E-06	0.26290335E-01
0.92801412E-06	0.64113589E-10	0.94458210E-06
0.64569294E-10	0.23462082E 03	0.62169797E-10
0.23028836E 03	0.47728159E-01	0.23250787E 03
0.47338188E-01	0.21609903E-05	0.47283698E-01
0.21471496E-05	0.15220304E-09	0.21349842E-05
0.15137572E-09	0.58783569E 03	0.14881628E-09
0.58931641E 03	0.85619926E-01	0.55707983E 03
0.85926175E-01	0.48433403E-05	0.81760764E-01
0.48035499E-05	0.47771065E-10	0.48296279E-05
0.23412466E-10	0.10753997E 04	0.93552666E-10
0.10753142E 04	0.13378012E 00	0.10459164E 04
0.13171929E 00	0.44058988E-05	0.13311303E 00
0.45075294E-05	0.75881812E-09	0.61718356E-05
0.54206484E-09	0.0	0.68265704E-09
0.16149600E 04	0.0	0.13580984E 04
0.17314088E 00	0.0	0.16377037E 00
0.18058653E-03	0.0	0.99265362E-05
-0.56019138E-06	0.0	0.26510321E-08
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

SECOND BALLISTIC TABLE

0.15000000E 02	0.20000000E 02	0.25000000E 02
0.50000000E 01	0.50000000E 01	0.50000000E 01
0.0	0.0	0.0
0.66000000E 04	0.67000000E 04	0.67000000E 04
0.11700000E 05	0.11600000E 05	0.11500000E 05
0.16100000E 05	0.15800000E 05	0.15500000E 05
0.18300000E 05	0.18000000E 05	0.17700000E 05
-0.31554557E-01	-0.31491395E-01	-0.22941940E-01
0.25779448E-01	0.25082666E-01	0.24186213E-01
0.94695156E-06	0.93580064E-06	0.91914688E-06
0.60011260E-10	0.58165986E-10	0.55651733E-10
0.22862090E 03	0.22753232E 03	0.22002965E 03
0.46527773E-01	0.45864277E-01	0.44413030E-01
0.20808957E-05	0.20489497E-05	0.19700592E-05
0.14860645E-09	0.15022479E-09	0.15091257E-09
0.53980225E 03	0.51918066E 03	0.49532886E 03
0.79564214E-01	0.77037036E-01	0.74122190E-01
0.47447866E-05	0.45918487E-05	0.43956334E-05
0.15901787E-09	0.22402442E-09	0.28825475E-09
0.99531567E 03	0.94033594E 03	0.88057104E 03
0.13141870E 00	0.12864327E 00	0.12473357E 00
0.65259246E-05	0.66923176E-05	0.65690820E-05
0.10109917E-08	0.14400028E-08	0.19796815E-08
0.13268040E 04	0.12711013E 04	0.12079033E 04
0.17620784E 00	0.18138206E 00	0.18635255E 00
0.12080532E-04	0.13822636E-04	0.14614283E-04
0.51119144E-08	0.92249621E-08	0.16653708E-07
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

SECOND BALLISTIC TABLE

0.30000000E 02	0.35000000E 02	0.40000000E 02
0.60000000E 01	0.60000000E 01	0.60000000E 01
0.0	0.0	0.0
0.68000000E 04	0.68000000E 04	0.69000000E 04
0.11500000E 05	0.11300000E 05	0.11300000E 05
0.15100000E 05	0.14800000E 05	0.14500000E 05
0.16900000E 05	0.16800000E 05	0.16400000E 05
0.17900000E 05	0.17800000E 05	0.17300000E 05
-0.28644178E-01	-0.21841597E-01	-0.28780818E-01
0.23122858E-01	0.21869406E-01	0.20466059E-01
0.88734464E-06	0.85127652E-06	0.80038455E-06
0.53501176E-10	0.50601703E-10	0.47954404E-10
0.21506778E 03	0.20397606E 03	0.19508344E 03
0.43050058E-01	0.40880807E-01	0.38795494E-01
0.19000763E-05	0.18118699E-05	0.17169932E-05
0.15483695E-09	0.15197599E-09	0.15403537E-09
0.47548755E 03	0.43850122E 03	0.41217017E 03
0.71608245E-01	0.66894293E-01	0.63341439E-01
0.42915708E-05	0.39733904E-05	0.38428825E-05
0.34277914E-09	0.38661874E-09	0.42076853E-09
0.80490918E 03	0.73785181E 03	0.66797239E 03
0.11687243E 00	0.11056018E 00	0.10242707E 00
0.74256477E-05	0.64986916E-05	0.63576917E-05
0.18582860E-08	0.25061468E-08	0.26545504E-08
0.10502024E 04	0.10050500E 04	0.90379736E 03
0.16334862E 00	0.17126489E 00	0.15902948E 00
0.15848927E-04	0.14069570E-04	0.16356615E-04
0.99359916E-08	0.23502199E-07	0.22665120E-07
0.12393630E 04	0.12140000E 04	0.10767466E 04
0.22717845E 00	0.26776493E 00	0.25129491E 00
0.42509957E-04	0.21551561E-03	0.46779285E-04
0.53700973E-07	-0.21170064E-06	0.24917728E-06

SECOND BALLISTIC TABLE

0.45000000E 02	0.50000000E 02	0.55000000E 02
0.60000000E 01	0.60000000E 01	0.60000000E 01
0.0	0.0	0.0
0.69000000E 04	0.70000000E 04	0.71000000E 04
0.11300000E 05	0.11300000E 05	0.11400000E 05
0.14300000E 05	0.14100000E 05	0.14100000E 05
0.16000000E 05	0.15900000E 05	0.15600000E 05
0.16900000E 05	0.16700000E 05	0.16400000E 05
-0.20488478E-01	-0.22628419E-01	-0.29156726E-01
0.18889520E-01	0.17182510E-01	0.15346885E-01
0.74784754E-06	0.68188183E-06	0.60863249E-06
0.44432485E-10	0.41117207E-10	0.37417167E-10
0.18052991E 03	0.16778123E 03	0.15302098E 03
0.35984889E-01	0.33202246E-01	0.30080359E-01
0.15812593E-05	0.14576026E-05	0.13108693E-05
0.15193713E-09	0.15019665E-09	0.14746657E-09
0.38244409E 03	0.34946387E 03	0.31834912E 03
0.59262638E-01	0.54629341E-01	0.50183192E-01
0.36411075E-05	0.34203986E-05	0.31512909E-05
0.45449089E-09	0.47187854E-09	0.51609961E-09
0.60533374E 03	0.53955396E 03	0.48698462E 03
0.94732404E-01	0.86631000E-01	0.79878032E-01
0.65658924E-05	0.53337844E-05	0.60712609E-05
0.27344842E-08	0.32452279E-08	0.34064707E-08
0.79880640E 03	0.73175977E 03	0.63196631E 03
0.14392483E 00	0.14196527E 00	0.12468064E 00
0.14867687E-04	0.14126343E-04	0.12899101E-04
0.21610344E-07	0.35084110E-07	0.31270943E-07
0.95618970E 03	0.87240991E 03	0.75603979E 03
0.23028231E 00	0.24170452E 00	0.21613413E 00
0.44658402E-04	0.51615963E-04	0.36492085E-04
0.24580714E-06	0.51307580E-06	0.50168404E-06

SECOND BALLISTIC TABLE

0.60000000E 02	0.65000000E 02	0.70000000E 02
0.60000000E 01	0.60000000E 01	0.60000000E 01
0.0	0.0	0.0
0.72000000E 04	0.75000000E 04	0.77000000E 04
0.11500000E 05	0.11700000E 05	0.12000000E 05
0.14000000E 05	0.14100000E 05	0.14400000E 05
0.15500000E 05	0.15400000E 05	0.15300000E 05
0.16200000E 05	0.16000000E 05	0.15800000E 05
-0.24174605E-01	-0.32901440E-01	-0.31744126E-01
0.13385557E-01	0.11338942E-01	0.91918670E-02
0.53091730E-06	0.44158674E-06	0.35281141E-06
0.33232861E-10	0.29298231E-10	0.24501304E-10
0.13628899E 03	0.12222952E 03	0.10286295E 03
0.26627243E-01	0.23337778E-01	0.19412529E-01
0.11473576E-05	0.10045978E-05	0.80708725E-06
0.14174609E-09	0.13903680E-09	0.13084864E-09
0.28329688E 03	0.24829070E 03	0.21187293E 03
0.45000017E-01	0.39804205E-01	0.34422044E-01
0.28866361E-05	0.25258360E-05	0.19856607E-05
0.52960347E-09	0.58544660E-09	0.71619755E-09
0.42210547E 03	0.36647168E 03	0.31567993E 03
0.70801616E-01	0.63388050E-01	0.57450302E-01
0.52035830E-05	0.52556379E-05	0.69982716E-05
0.36956218E-08	0.42782418E-08	0.53127280E-08
0.55252026E 03	0.46718872E 03	0.37693311E 03
0.11524916E 00	0.10171026E 00	0.84457276E-01
0.12569270E-04	0.14559499E-04	0.16544145E-04
0.39797975E-07	0.41944464E-07	0.36203026E-07
0.65304980E 03	0.54254004E 03	0.42782983E 03
0.20639485E 00	0.18000495E 00	0.13581479E 00
-0.41450883E-04	-0.81888022E-04	-0.14482073E-04
0.89489060E-06	0.97808334E-06	0.53329347E-06

SECOND BALLISTIC TABLE

```
 0.75000000E 02         0.80000000E 02
 0.50000000E 01         0.50000000E 01
 0.0                    0.0
 0.81000000E 04         0.81000000E 04
 0.12500000E 05         0.12800000E 05
 0.14600000E 05         0.14700000E 05
 0.15700000E 05         0.15600000E 05
-0.36948875E-01        -0.14113124E-01
 0.69814064E-02         0.46739683E-02
 0.25841609E-06         0.17448326E-06
 0.19570623E-10         0.13162678E-10
 0.83880356E 02         0.56320923E 02
 0.15458006E-01         0.10449842E-01
 0.61404018E-06         0.37701369E-06
 0.12338225E-09         0.90918648E-10
 0.17431053E 03         0.12323389E 03
 0.28800629E-01         0.20682942E-01
 0.16861541E-05         0.12522632E-05
 0.85933771E-09         0.79211659E-09
 0.25014171E 03         0.17248473E 03
 0.50448716E-01         0.35895605E-01
-0.30938656E-06         0.13514054E-05
 0.12946110E-07         0.11027577E-07
 0.32262988E 03         0.21397000E 03
 0.11146045E 00         0.73082030E-01
-0.71440489E-04        -0.37487727E-04
 0.68820083E-06         0.41663935E-06
 0.0                    0.0
 0.0                    0.0
 0.0                    0.0
 0.0                    0.0
 0.0                    0.0
```

THIRD BALLISTIC TABLE

0.50000000E 01	0.12500000E 02	0.20000000E 02
0.20000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.10900000E 05	0.12100000E 05	0.13200000E 05
-0.44590539E 00	-0.53771842E 00	-0.44000703E 00
0.80811024E-01	0.80891967E-01	0.80717921E-01
-0.24198471E-05	-0.24224100E-05	-0.23536995E-05
0.39302381E-10	0.51937607E-10	0.56261590E-10
0.64352954E 03	0.71536743E 03	0.78840942E 03
0.44786330E-01	0.48332345E-01	0.48795857E-01
-0.15720543E-05	-0.24149340E-05	-0.30826341E-05
0.65199984E-10	0.14911321E-09	0.29150904E-09

THIRD BALLISTIC TABLE

0.27500000E 02	0.35000000E 02	0.42500000E 02
0.20000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.13100000E 05	0.12500000E 05	0.12500000E 05
-0.23965603E 00	-0.10349798E 00	0.58434587E-01
0.80394506E-01	0.80158770E-01	0.79893589E-01
-0.22453005E-05	-0.21576352E-05	-0.20636116E-05
0.60657895E-10	0.62984548E-10	0.63653652E-10
0.80331128E 03	0.78743481E 03	0.83023071E 03
0.51646613E-01	0.55954114E-01	0.55985902E-01
-0.32957878E-05	-0.32531534E-05	-0.25534246E-05
0.32480907E-09	0.26014990E-09	0.15170151E-09

THIRD BALLISTIC TABLE

0.50000000E 02	0.57500000E 02	0.65000000E 02
0.20000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.12600000E 05	0.12300000E 05	0.12100000E 05
0.26067120E 00	0.31261128E 00	0.36220652E 00
0.79583824E-01	0.79466760E-01	0.79360187E-01
-0.19605204E-05	-0.19073050E-05	-0.18607052E-05
0.62568561E-10	0.63513209E-10	0.63888755E-10
0.81608350E 03	0.80682739E 03	0.80098901E 03
0.55133119E-01	0.56200292E-01	0.56569297E-01
-0.15843616E-05	-0.88705434E-06	-0.12379900E-06
-0.32985684E-10	-0.20511468E-09	-0.38566927E-09

THIRD BALLISTIC TABLE

```
 0.72500000E 02         0.80000000E 02
 0.20000000E 01         0.20000000E 01
 0.0                    0.0
 0.11900000E 05         0.11800000E 05
 0.38043833E 00         0.41003025E 00
 0.79310536E-01         0.79264224E-01
-0.18331775E-05        -0.18121900E-05
 0.64555070E-10         0.64770703E-10
 0.79335693E 03         0.78961401E 03
 0.56353331E-01         0.57576522E-01
 0.90817429E-06         0.48777076E-06
-0.60241212E-09        -0.54249538E-09
```

FOURTH BALLISTIC TABLE

0.50000000E 01	0.30000000E 02	0.55000000E 02
0.20000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.13000000E 05	0.13000000E 05	0.13000000E 05
-0.28341883E-02	-0.37907360E-02	-0.60301088E-02
0.50878822E-04	0.52425778E-04	0.55562079E-04
0.11742525E-09	-0.18399496E-09	-0.97077346E-09
0.65568590E-13	0.94545845E-13	0.16145548E-12
0.82518965E 00	0.85653239E 00	0.91206771E 00
0.97729615E-04	0.13786182E-03	0.16013747E-03
-0.83450669E-09	-0.10282797E-07	-0.28855379E-07
0.32401886E-12	0.19301071E-11	0.91562374E-11

FOURTH BALLISTIC TABLE

```
 0.80000000E 02
 0.20000000E 01
 0.0
 0.12000000E 05
-0.74683204E-02
 0.57530691E-04
-0.14748769E-08
 0.20581860E-12
 0.94437510E 00
 0.18925199E-03
-0.58251516E-07
 0.22669838E-10
```

FIFTH BALLISTIC TABLE

0.50000000E 01	0.15000000E 02	0.25000000E 02
0.20000000E 01	0.30000000E 01	0.30000000E 01
0.0	0.0	0.0
0.12100000E 05	0.12800000E 05	0.13100000E 05
0.22236040E 00	0.19300000E 05	0.18500000E 05
-0.53894799E-03	0.15188676E 00	-0.93585432E-01
0.40779814E-05	-0.21471674E-03	0.19803554E-03
-0.36913653E-10	0.39492988E-05	0.38012631E-05
0.52364282E 03	-0.33552883E-10	-0.28371888E-10
0.84621072E-01	0.57215796E 03	0.58920825E 03
-0.46131991E-05	0.84626019E-01	0.82485080E-01
0.21487628E-09	-0.55600822E-05	-0.53406056E-05
0.0	0.41261439E-09	0.58536398E-09
0.0	0.10013298E 04	0.97171997E 03
0.0	0.67116618E-01	0.79683363E-01
0.0	0.55005166E-05	0.74994305E-05
0.0	0.33316916E-08	0.66678254E-08

FIFTH BALLISTIC TABLE

0.35000000E 02	0.45000000E 02	0.55000000E 02
0.30000000E 01	0.30000000E 01	0.30000000E 01
0.0	0.0	0.0
0.12900000E 05	0.11300000E 05	0.10800000E 05
0.17800000E 05	0.15100000E 05	0.14700000E 05
-0.41843438E 00	-0.48467010E 00	-0.49684834E 00
0.69065648E-03	0.84353215E-03	0.90644206E-03
0.36389201E-05	0.35661506E-05	0.35226440E-05
-0.21692620E-10	-0.20071875E-10	-0.19723043E-10
0.56617749E 03	0.43536328E 03	0.39534106E 03
0.81950307E-01	0.78123868E-01	0.73956490E-01
-0.45800498E-05	0.13425802E-06	0.13913241E-05
0.66831674E-09	-0.12577402E-09	-0.18058495E-09
0.93731982E 03	0.72739648E 03	0.69429565E 03
0.89232802E-01	0.79372883E-01	0.83643973E-01
0.47504655E-04	-0.43911068E-05	-0.64221495E-05
-0.88343370E-07	0.32075611E-08	0.48752788E-08

FIFTH BALLISTIC TABLE

```
  0.65000000E 02        0.77500000E 02
  0.30000000E 01        0.30000000E 01
  0.0                   0.0
  0.10400000E 05        0.12400000E 05
  0.14500000E 05        0.14400000E 05
 -0.48594517E 00       -0.49406999E 00
  0.92464546E-03        0.95620775E-03
  0.34981940E-05        0.34736058E-05
 -0.20057053E-10       -0.19873145E-10
  0.36494556E 03        0.34957910E 03
  0.70768952E-01        0.69327225E-01
  0.19479094E-05        0.18963892E-05
 -0.15469485E-09       -0.72263792E-10
  0.67706396E 03        0.66901294E 03
  0.89440107E-01        0.90105951E-01
 -0.12786430E-04       -0.10780218E-04
  0.92809280E-08        0.11096141E-07
```

```
****************************************************************************************************************************************
****************************************************************************************************************************************
****************************************************************************************************************************************
```

SIXTH BALLISTIC TABLE

0.50000000E 01	0.30000000E 02	0.55000000E 02
0.20000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.12500000E 05	0.12500000E 05	0.12500000E 05
-0.40031224E-02	-0.34229141E-02	-0.56032427E-02
0.46734667E-05	0.38393910E-05	0.67141145E-05
-0.11574185E-09	0.12075130E-09	-0.62236061E-09
0.15426164E-12	0.14128960E-12	0.19526080E-12
0.32573949E 00	0.33625793E 00	0.36970794E 00
0.15521410E-03	0.14298031E-03	0.15486017E-03
-0.52424767E-08	-0.54800893E-08	-0.28209815E-07
0.42974982E-12	0.11090616E-11	0.78686432E-11

SIXTH BALLISTIC TABLE

```
 0.80000000E 02
 0.20000000E 01
 0.0
 0.12500000E 05
-0.69053061E-02
 0.86051368E-05
-0.11421217E-08
 0.23524195E-12
 0.38881576E 00
 0.16382609E-03
-0.46488416E-07
 0.16285445E-10
```

SEVENTH BALLISTIC TABLE

0.50000000E 01	0.12500000E 02	0.20000000E 02
0.20000000E 01	0.30000000E 01	0.30000000E 01
0.0	0.0	0.0
0.12000000E 05	0.12300000E 05	0.12500000E 05
-0.10400724E 00	0.15700000E 05	0.16000000E 05
0.81964128E-04	-0.19927856E-01	0.47894921E-01
-0.24321980E-05	-0.74988275E-04	-0.21399414E-03
0.38499468E-10	-0.23680141E-05	-0.21303622E-05
-0.27935596E 03	0.34406769E-10	0.30822858E-10
-0.46370249E-01	-0.29312134E 03	-0.30100195E 03
0.12425826E-04	-0.41252948E-01	-0.42060789E-01
-0.31882696E-09	0.87435765E-05	0.10397433E-04
0.0	0.45569370E-09	0.20758506E-09
0.0	-0.31476685E 03	-0.31246606E 03
0.0	0.30210268E-01	0.34307875E-01
0.0	0.10515914E-04	0.90497397E-05
0.0	-0.86408836E-09	-0.12371300E-08

SEVENTH BALLISTIC TABLE

0.27500000E 02	0.35000000E 02	0.42500000E 02
0.30000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.12500000E 05	0.12400000E 05	0.12300000E 05
0.16000000E 05	0.25243294E 00	0.29942024E 00
0.15360188E 00	-0.54136757E-03	-0.63120294E-03
-0.38518966E-03	-0.21907708E-05	-0.21524647E-05
-0.22474687E-05	0.23537020E-10	0.21259883E-10
0.26968414E-10	-0.29566284E 03	-0.29190332E 03
-0.30073047E 03	-0.45320768E-01	-0.40129982E-01
-0.43030698E-01	0.13632462E-04	0.10684284E-04
0.11562372E-04	-0.66856698E-09	-0.34815195E-09
-0.84903848E-10	0.0	0.0
-0.31397021E 03	0.0	0.0
0.31575285E-01	0.0	0.0
0.66508674E-05	0.0	0.0
-0.96373221E-09	0.0	0.0

SEVENTH BALLISTIC TABLE

0.50000000E 02	0.57500000E 02	0.65000000E 02
0.30000000E 01	0.30000000E 01	0.30000000E 01
0.0	0.0	0.0
0.12100000E 05	0.11900000E 05	0.11800000E 05
0.16700000E 05	0.16300000E 05	0.16000000E 05
0.37056339E 00	0.43011218E 00	0.43593097E 00
-0.74601476E-03	-0.83902106E-03	-0.86247409E-03
-0.21101450E-05	-0.20754605E-05	-0.20603393E-05
0.18700735E-10	0.16589480E-10	0.15708629E-10
-0.28313037E 03	-0.27464209E 03	-0.27008325E 03
-0.39473955E-01	-0.38341399E-01	-0.36503717E-01
0.91799875E-05	0.73656047E-05	0.59303948E-05
-0.19817599E-09	0.24281105E-10	0.20906725E-09
-0.28869995E 03	-0.29744995E 03	-0.30194995E 03
0.41098051E-01	0.39064229E-01	0.38880158E-01
0.12018553E-04	0.40233144E-05	-0.18969629E-04
0.79958852E-07	0.14328140E-06	0.26159921E-06

SEVENTH BALLISTIC TABLE

0.72500000E 02	0.80000000E 02
0.30000000E 01	0.30000000E 01
0.0	0.0
0.11800000E 05	0.11700000E 05
0.15800000E 05	0.15600000E 05
0.39600939E 00	0.42041957E 00
-0.82055433E-03	-0.85407309E-03
-0.20642947E-05	-0.20510815E-05
0.16056295E-10	0.15222337E-10
-0.26939136E 03	-0.26509839E 03
-0.34117918E-01	-0.34522444E-01
0.47504645E-05	0.44801745E-05
0.38675019E-09	0.39979131E-09
-0.30364990E 03	-0.30666992E 03
0.48963636E-01	0.32865789E-01
-0.16947088E-03	-0.21991495E-04
0.83826842E-06	0.27331424E-06

EIGHTH BALLISTIC TABLE

0.50000000E 01	0.30000000E 02	0.55000000E 02
0.20000000E 01	0.20000000E 01	0.20000000E 01
0.0	0.0	0.0
0.12600000E 05	0.12700000E 05	0.11900000E 05
0.78984350E-02	0.60268939E-02	0.43568313E-02
-0.94776651E-05	-0.68760019E-05	-0.50171911E-05
0.16801243E-08	0.98311603E-09	0.59067928E-09
-0.19521174E-12	-0.14947207E-12	-0.13104155E-12
-0.23902911E 00	-0.23778242E 00	-0.20235212E 00
-0.20282694E-03	-0.17015025E-03	-0.12539959E-03
0.14314381E-07	0.64431980E-08	0.60069745E-08
-0.42029764E-12	0.15314260E-11	0.34577972E-12

EIGHTH BALLISTIC TABLE

```
 0.80C0C0Q0E 02
 0.2COCCOCOE C1
 0.C
 0.1130C000E C5
 C.29C51665E-C2
-0.32701892E-C5
 0.20453642F-09
-0.11043857E-12
-0.16838163E CC
-0.11204566E-C3
 0.84507228E-C8
-0.96344634E-12
```

NINTH BALLISTIC TABLE

0.50CCC000E C1	0.30C00000E 02	C.55CCCCC0E 02
0.2CCCCCC0E C1	0.20000000E 01	0.200CCCC0E 01
0.C	0.C	C.0
0.12C00000E C5	0.12000000E 05	C.135CCCCCE 05
C.1CCC5646E C1	0.99729544E 00	0.980CC5C3E 00
0.39535269E-04	0.5C992487E-C4	0.78381C25F-04
0.15712018E-08	-0.69385650E-10	-0.59941350E-08
0.33104836E-13	0.1811C844E-12	0.64646118E-12
0.17551136E 01	0.18885870E C1	0.243925CCE 01
0.972C1966E-C4	0.22189190E-03	0.11032CCCE-02
-0.13142813E-C7	-0.44501C11E-C7	-0.830428C6E-06
0.60034481E-12	0.74270624E-11	0.28619240E-09
0.C	0.0	0.0
C.C	0.C	0.0
0.0	0.0	0.0
C.C	0.0	0.0
0.0	0.0	0.0

NINTH BALLISTIC TABLE

```
 0.80000000E 02
 0.30000000E 01
 0.0
 0.12000000E 05
 0.15200000E 05
 0.98125958E 00
 0.79034871E-04
-0.61084968E-08
 0.74429899E-12
 0.23181391E 01
 0.58374787E-03
-0.27691374E-06
 0.11478704E-09
 0.51725197E 01
 0.57250932E-02
-0.14298297E-04
 0.35073608E-07
```

TENTH BALLISTIC TABLE

0.0	0.30000000E 02	0.80000000E 02
0.10000000E 01	0.10000000E 01	0.10000000E 01
0.0	0.0	0.0
0.84499997E 00	0.88299996E 00	0.94699997E 00
0.0	0.0	0.0
0.10000000E 01	0.99040598E 00	0.97441596E 00
0.0	0.51994499E-06	0.13866511E-05
0.10015929E 00	0.10010748E 01	0.10000000E 01
-0.10728629E-04	-0.72468247E-05	0.0

ELEVENTH BALLISTIC TABLE

0.50000000E 01	0.20000000E 02	0.40000000E 02
0.20000000E 01	0.20000000E 01	0.30000000E 01
0.0	0.0	0.0
0.15000000E 05	0.15000000E 05	0.14000000E 05
0.44295695E-01	0.39856613E-01	0.16500000E 05
0.11021075E-05	0.13935942E-05	0.30672155E-01
0.61905224E-01	0.61612882E-01	0.14965781E-05
0.20288544E-05	0.32180051E-05	0.51941060E-01
0.0	0.0	0.42543634E-05
0.0	0.0	0.62320661E-01
0.0	0.0	0.74444060E-05

ELEVENTH BALLISTIC TABLE

0.60000000E 02	0.80000000E 02
0.30000000E 01	0.30000000E 01
0.0	0.0
0.14000000E 05	0.14000000E 05
0.16000000E 05	0.15600000E 05
0.17343938E-01	0.51356144E-02
0.13595636E-05	0.58914418E-06
0.26460474E-01	0.13199501E-01
0.54652728E-05	0.35339954E-05
0.48195675E-01	0.19671310E-01
0.15476704E-04	0.15286496E-04

REFERENCES

1. Gun Fire Control System MK 86 Mod 3 Computer Program Design Specification, Volume 2, prepared for Naval Ordnance Systems Command by Lockheed Electronics Company, Inc., November, 1971.

2. Mark 86 Mod 3-Load 07 S/N 4, M86-0004-03-0700 Part I, II prepared by Lockheed Electronics Company, Inc.

3. Gun Fire Control System MK 86 Mod 3 Computer Subprogram Design Specification, prepared for Naval Ordnance Systems Command by Lockheed Electronics Company, Inc., April, 1972.

4. Gun Fire Control System MK 86 Mod 3 Computer Program Performance Specification, Volume 2, prepared for Naval Ordnance Systems Command by Lockheed Electronics Company, Inc., November, 1972.

5. Verifications of GFCS MK 86 Ballistics/Gun Order Computations, report prepared by Lockheed Electronics Company, Inc., August, 1972.

DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor J. R. Ward, Code 52Wa Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor D. E. Kirk, Code 52Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Professor H. A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	LT D. F. Regener, USN Naval Ship Missile System Engineering Station Space City 8 Port Hueneme, California 93043	2
7.	Mr. Spencer Greenwald Mr. Theodore Dougherty Naval Weapons Laboratory Dahlgren, Virginia 22448	2
8.	LT T. J. Will, USN SMC #1964 Naval Postgraduate School Monterey, California 93940	1
9.	LT D. R. Kidd, USN 1280 Hilby Avenue Seaside, California 93955	1

10. LT L. S. Beers, USN 1
SMC #1165
Naval Postgraduate School
Monterey, California 93940

11. LT J. D. Gorman, USN 1
36 Legare Street
Charleston, South Carolina 29401

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE Description and FORTRAN Simulation of MK 86 AA Ballistics	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Master's Thesis; June 1973	
5. AUTHOR(S) (First name, middle initial, last name) Delbert R. Kidd	

6. REPORT DATE June 1973	7a. TOTAL NO. OF PAGES 127	7b. NO. OF REFS 5
8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	

13. ABSTRACT

The GFCS MK86 solves the ballistics problem for the 5"/54 gun iteratively utilizing table look-ups. Each ballistics parameter table is sub-divided two-dimensionally, corresponding to selected elevation and slant range intervals. The final value for each parameter is found by interpolating between parameter values evaluated at elevation angles which bracket the actual elevation angle.

This thesis describes the method of determining the ballistics solution for AA cartesian ballistics. A description of the ballistics geometry and of the model utilized is presented, together with a FORTRAN simulation of the MK86 computer program and a listing of the ballistics tables used in this system.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Mark 86 Gun Fire Control System						
Ballistics						

Thesis
K3967 Kidd
c.1

144065

Description and FORTRAN simulation of MK 86 AA ballistics.